2016

Web Based Bioinformatics Tool for Contributing Content to the Anti-Microbial Peptide Editable Database

Abraham Herrera
University of Rhode Island, abraham7_12@yahoo.com

Follow this and additional works at: <https://digitalcommons.uri.edu/theses>

Recommended Citation

Herrera, Abraham, "Web Based Bioinformatics Tool for Contributing Content to the Anti-Microbial Peptide Editable Database" (2016). *Open Access Master's Theses*. Paper 1133.
<https://digitalcommons.uri.edu/theses/1133>

This Thesis is brought to you for free and open access by DigitalCommons@URI. It has been accepted for inclusion in Open Access Master's Theses by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons-group@uri.edu.

**WEB BASED BIOINFORMATICS TOOL FOR CONTRIBUTING CONTENT TO THE ANTI-
MICROBIAL PEPTIDE EDITABLE DATABASE**

BY

ABRAHAM HERRERA

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER SCIENCE

UNIVERSITY OF RHODE ISLAND

2016

MASTER OF SCIENCE THESIS IN COMPUTER SCIENCE
OF
ABRAHAM HERRERA

APPROVED:

Thesis Committee:

Major Professor: Joan Peckham

Lenore M. Martin

Lisa DiPippo

Roberta King

Nasser H. Zawia
DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2016

ABSTRACT

The goal of the work described in this thesis is to improve the software tool used to extract relevant data from an existing larger database, the Protein Data Bank (PDB), in order to populate the newly designed AMPed database that has been developed by Professor Lenore Martin's microbial peptide research group at URI. However, not all data from the PDB site can be automatically loaded into the AMPed database. Thus, this work describes a user-friendly secure web-based tool for manually transferring, editing, and contributing structural and anti-microbial experimental data generated by research labs around the globe into the AMPed.

Once users have uploaded their data into AMPed, it then goes through an evaluation process where it is placed in a 'pending' mode, waiting to be reviewed by Professor Martin or other designated evaluator ('reviewer'). The reviewer gets notified through email when a new entry has been submitted or added to the database.

The designated content reviewer can access the newly uploaded data through a private and secure web-based user interface to evaluate the incoming entries. Within this tool, the reviewer can approve the data, edit it, or simply reject it. The reviewer can then communicate back to the data contributors through email, and inform them about the status of their contributions. If the data has been approved for addition to AMPed, it will rapidly be made available to anyone searching the database.

In addition, we have increased the efficiency of the AMPed website through major improvements in graphical design and usability using the latest techniques available.

ACKNOWLEDGMENTS

This project would not have been possible if it was not for the commitment, support, and guidance of my professors Dr. Joan Peckham and Dr. Lenore M. Martin; their encouragement and dedication led me to be part of this magnificent project. For all those moments that you both spent with me sharing your knowledge and expertise and for pushing me to exceed the expectations, I will always be thankful.

I would also like to extend my gratitude to Terry Wild and Chi Shen for helping us with the creation of the virtual server and moving the AMPed site to a faster, newer, more reliable and secure server.

To my bioinformatics class and team mates (Brett Boudreau, Christopher Mills, Mingyang Zhao, and Vivek Annavarjula) who helped me to map the AMPed and PDB tables and setting the floor plan for populating the AMPed database.

Special thanks go to my friend and predecessor Tripti Garg for her support during this project, as well as to my kids and friends for believing in me, for their support and admiration.

Finally, I would also like to extend a special gratitude to my wife, Zoila Castro, for her continuous encouragement throughout this entire project and in my life.

Table of Contents

ABSTRACT.....	ii
ACKNOWLEDGMENTS.....	iii
Table of Contents.....	iv
List of Tables	viii
LIST OF FIGURES.....	ix
CHAPTER 1.....	1
INTRODUCTION	1
CHAPTER 2.....	5
AMPed Domain: A WEB BASED BIOINFORMATICS TOOL	5
AMPed Domain	6
The Hosting Server	6
AMPed Web Flow	8
Secure Server	9
Portability.....	12
Branding.....	18
Visitor Counter	19
The Header and Footer Sections.....	21

CHAPTER 3.....	27
THE AMPED DATABASE CONTENT	27
Importing content to AMPed.....	27
The BioParser	29
Matching Tables: PDB – AMPed	34
Entity Relationship (ER) Diagram	34
The Tables	36
AMPed Content.....	40
Navigational Features	41
CHAPTER 4.....	44
JOIN US & LOGIN PAGES	44
Join Us Page	44
Validation Process.....	46
Registration Process.....	48
Login Page	50
Access levels.....	52
CAPTCHA	53
CHAPTER 5.....	57
ADMINISTRATOR SITE	57

User Portal	59
Peptide Portal	61
CHAPTER 6.....	63
SEARCH SECTION.....	63
Search Page.....	63
Search Criteria	68
Advanced Search	68
Chapter 7.....	71
ABOUT US.....	71
CHAPTER 8.....	72
CONTACT US.....	72
Chapter 9.....	75
CONCLUSION AND FUTURE WORK	75
GLOSSARY.....	79
APPENDIX A.....	83
Chart matching the AMPed and PDB tables	83
APPENDIX B.....	87
Database Structure and Schema of AMPed.....	87
APPENDIX C.....	95

Comparison between AMPed and other antimicrobial peptides repositories online	95
APPENDIX D.....	99
Perl code to read and parse a PDBML/XML file and output run	99
APPENDIX E	108
ER Diagram.....	108
BIBLIOGRAPHY	110

List of Tables

Table 1: AMPed tables with attributes and data type.....	90
Table 2: AMPed comparison with other Database repositories	98

LIST OF FIGURES

Figure 1: AMPed Web Flow Diagrams	9
Figure 2: HTTP vs HTTPS.....	10
Figure 3: Secure icon.....	11
Figure 4: AMPed's Certificate Information	11
Figure 5: Bootstrap's grid system	14
Figure 6: AMPed index page HTML source code	15
Figure 7: Desktop's AMPed index homepage full-width display	16
Figure 8: Mobile's AMPed index homepage truncated width display.....	17
Figure 9: AMPed logo.....	18
Figure 10: favicon - shortcut icon	18
Figure 11: Google Analytics - Geo report of AMPed.....	21
Figure 12: Google Analytics code inserted on the footer section of the AMPed's pages	21
Figure 13: AMPed header before login.....	22
Figure 14: AMPed header after login.....	23
Figure 15: AMPed header with logo reference on intermediate pages.....	23
Figure 16: AMPed footer	24
Figure 17: AMPed copyright info	24
Figure 18: ER diagram of AMPed database	35
Figure 19: Tables of AMPed database	39
Figure 20: Peptide Database table showing AA Sequences	40
Figure 21: Browsing the tables	41

Figure 22: "Join Us" link at header section	44
Figure 23: Join Us page fields.....	45
Figure 24: Welcome to AMPed registration email	49
Figure 25: Login page	51
Figure 26: Welcome message header.....	52
Figure 27: AMPed CAPTCHA image	54
Figure 28: Secure access to AMPed Admin Tools	57
Figure 29: AMPed - User Portal snapshot	59
Figure 30: User's Portal pop-up window	60
Figure 31: Peptide contribution by user	62
Figure 32: AMPed Search page	64
Figure 33: AJAX code to display Peptides results as the entry is input.....	65
Figure 34: Live search with a list of possible matching results.....	66
Figure 35: Live search without a matching hit	67
Figure 36: Full vs. Partial match.....	67
Figure 37: Advance Search page	69
Figure 38: About Us page.....	71
Figure 39: Contact Us page	73
Figure 40: AMPed structure version 1.0	92
Figure 41: AMPed structure version 2.0	93
Figure 42: AMPed structure version 3.0	94
Figure 43: AMPed ER diagram (High Definition).....	109

CHAPTER 1

INTRODUCTION

Antibiotic-resistant bacteria pose a very serious threat to the medical community, as antibiotics are essential for carrying out many conventional therapeutic techniques such as surgery. Antimicrobial peptides (AMPs), also called host defense peptides (HDPs), are an essential part of the innate immune response found in all domains of life, both in invertebrate and vertebrate organisms. These short chains of amino acids have been demonstrated to kill bacteria, fungi, and viruses, and even combat the growth of cancer cells *in vivo*[1]. For that reason, since the 1980s, there has been a large effort to better understand and mimic their antimicrobial action in order to help fight the emergence of antibiotic-resistant bacteria, viruses such as HIV-1, and cancer [2].

Over the past few years, the problem of how to share and access an exponentially-growing amount of peptide information has increased dramatically. This has led to a demand for new software tools, analytical methods, and easy-to-use software for biological users to collect and handle large amounts of new data.

The Anti-Microbial Peptide Editable Database (“AMPed”), developed at URI for carrying out research on the structure and functions of antimicrobial proteins of up to 100 amino acids in length, is a high-quality annotated collection of data describing the properties of antimicrobial peptides. It was specifically designed to provide a user-friendly resource for interdisciplinary researchers seeking to share data and information regarding AMPs.

For this thesis the following goals have been set:

1. Transfer the old code and run both the database (DB) and tools to a newer and more secure server.
2. Populate the AMPed DB with a variety of information types from existing databases.
3. Provide new tools to allow users to manually contribute new data to the database.
4. Enhance the overall user experience.

To reach these goals, a redesigned and updated version of the bioparser software tool written in Practical Extraction and Report Language (Perl) script was implemented to parse and import bulk data downloaded from the Protein Data Bank (PDB) into the newly redesigned AMPed database in a uniform and comprehensive way [3]. This addressed a pressing user need because existing protein databases, such as Swiss-Prot, NCBI, and PDB, with broader subject matter than just peptides, are often difficult and time-consuming to search effectively for AMPs.

Since not every type of experimental data can be parsed automatically for loading into the database, there was a demonstrated need for a mechanism for manually uploading and curating experimental data and antimicrobial peptide potency. This project implements a new data import capability, allowing biological researchers from around the world ('contributors') to upload their findings directly into the AMPed database, data which, while published in summary form, may not be found in any other online database.

A graphical user interface (GUI) was designed to collect all the needed information from potential contributors, and then allows them to input data into the appropriate tables of the AMPed database at URI.

Once the information has been uploaded into the AMPed database, it goes through a preliminary evaluation process, where it is placed in a 'pending' mode, to be reviewed by Professor Martin or another designated evaluator ('reviewer'). The 'reviewer' gets notified through email whenever new data has been uploaded into the database.

The designated content reviewer can then access the newly uploaded data through a private-secured web user interface to evaluate the incoming data entries. With this tool, the reviewer is able to either approve the data, edit it, or simply reject it. The reviewer then has the option of communicating back to the contributors through email and informing them about the status of their contributions. If the reviewer approves the submitted data, the contributors will be notified, and the data will be made available to anyone searching the AMPed database. If not, the reviewer has the opportunity to explain to the contributors the reasons that the entry was rejected and to request further information and/or missing data.

REFERENCES

- [1] A. Izadpanah, RL. Gallo, "*Antimicrobial peptides*". J Am Acad Dermatol. 2005 Mar;52(3 Pt 1):381-90; quiz 391-2 [Online]. Available:
<https://www.ncbi.nlm.nih.gov/pubmed/15761415>
- [2] Guangshun Wang, Xia Li and Zhe Wang, "*APD3: the antimicrobial peptide database as a tool for research and education*", Oxford Journals. Volume 44, Issue D1, Pp. D1087-D1093 [Online]. Available:
<http://nar.oxfordjournals.org/content/44/D1/D1087.full>
- [3] T. Garg. (2014). "*Graphical User Interface for Antimicrobial Peptide Database*" (Master's thesis). Retrieved from <http://digitalcommons.uri.edu/cgi/viewcontent.cgi?article=1836&context=theses>

CHAPTER 2

AMPed Domain: A WEB BASED BIOINFORMATICS TOOL

The Internet has become a significant instrument in our day-to-day activities - from an entertainment source to a research tool, from allowing us to pay bills online, make purchases, or perform any kind of transactions to “link people and information through computers and other digital devices.” [1] In this “information age” we live in, we are constantly exposed and bombarded with stimuli that can make us forget the primary objective of the World Wide Web: the support of group discussions, the exchange of electronic correspondence, and the access to distant databases. [2] Having the information available and accessible from a database is as important as being able to process this information and perform quick queries as well as being able to contribute to the repository.

The AMPed database has been in development for several years under the direction of Professor Lenore Martin. Its creation has been the work of dedicated students taking an unusual interdisciplinary project-oriented class, which pairs graduate students from Biology and Computer Sciences to collaborate together. David Ryder, a biochemist, and Daniel Ducharme, a computer scientist, initiated it as a class project for their CMB/CSC 522 class. George Konstantinidis then designed the first extensive database structure, known here as DBS version 1.0. [3] Tripti Garg, together with the assistance of her CMB/CSC 522 class, enhanced this structure on her thesis project, known here as DBS version 2.0. She also designed and implemented the first AMPed’s Web Interface,

known here as AWI version 1.0 [4]. In this project, with the help and guidance of Professor Martin and students in the CMB/CSC 522 classes, the database structure as well as the GUI interface has been enhanced; referenced here as DBS version 3.0 and AWI version 2.0 accordingly. The site is entirely written in PHP scripts with MySQL commands enhanced with Cascading Style Sheets (CSS) and Asynchronous JavaScript and XML (AJAX), a client-side script that communicates to and from a server/database.

AMPed Domain

Thanks to Ms. T. Wild and Chi Shen in the IT department, the AWI version 1.0 code was transferred to a newer virtualized server. That AWI version showed many errors due to the fact that it was created using a lower PHP version than the one installed on the new server. The entire AWI version 1.0 was revised and upgraded to comply with the newer PHP version. This enhancement and many others that will be explained in later chapters are part of the AMPed Web Interface version 2.0. One of these enhancements is the switch from an Internet Protocol (IP) address, a set of numerical instructions used by the computer to know the exact location of the pages, 131.128.169.79 on the old server, to a newly created and customized domain name amped.uri.edu with a new IP address, 131.128.20.101. A domain name is the readable version of the IP address.

The Hosting Server

The hosting server of the AMPed website version 1.0 was a 10+ years old Linux box locating in the Biochemistry Lab. This old server was too unstable for hosting a public

database and was frequently offline. During this project, the AMPed website was moved to a brand new virtualized Linux server running GNU/Linux.

The hosting server for the AMPed database and website went from a:

- Processor: Pentium 4 variant, 3.4Ghz (single core)
- Memory: 2 GB
- Disk space: 225 GB total

to a server with the below specifications:

- Processor: Pentium Xeon 3.3Ghz, 4 cores
- Memory: 8 GB
- Disk space: 1TB

The upload and download Internet speed for this server is about 500 Mbps download, 150 Mbps upload with a wide bandwidth. As a mode of comparison, the typical home internet service is about 25 Mbps download and 5 Mbps upload.

The following software is installed and running on the server:

- Perl 5.16.3
- PHP 5.6.27
- MySQL 5.5
- Apache 2.4.6
- Python 2.7.5

In short, the AMPed site is running on a faster, newer, more reliable and higher capacity Linux server. Running on a Linux server in a dedicated facility is more secure and stable than running on a lab-based server box. [13]

AMPed Web Flow

A web flow or web flowchart is a diagram showing a sequence of actions or movements performed by a user within the web site. On the AWI version 1.0 the flow chart displayed a few set of options from the homepage: a search option with detail results, the About Us page, and the logout and login options with the validation process. The diagram in figure 1 displays both the AWI version 1.0 and 2.0 flow charts. From the website's homepage users can access not only the options listed on the AWI version 1.0 web flow but many more pages depending on their access level (more information about the levels in Chapter 4, Access Levels section).

AWI version 1.0 flow chart

AWI version 2.0 flow chart

Figure 1: AMPed Web Flow Diagrams

Secure Server

On AWI version 1.0 the AMPed database was accessed through a regular Hyper Text

Transfer Protocol (HTTP), a procedure used to request and transmit files such as

webpages and webpage components, over the Internet or other computer network [5].

Considering that the AWI version 2.0 allows users to join the AMPed database by

entering personal information, such as full name, phone number, full address, and email

among other information, it was evident that the database needed to be accessed from

a more secure protocol. Working with the IT team, the Hyper Text Transfer Protocol

Secure (HTTPS) was adopted. This increases security for users' login credentials by encrypting all communications between the user's browser and the AMPed website.

Figure 2 shows a graphical representation of these two protocols.

Figure 2: HTTP vs HTTPS

“HTTPS pages typically use one of two secure protocols to encrypt communications - SSL (Secure Sockets Layer) or TLS (Transport Layer Security).” [6]

The <https://amped.uri.edu> secure website has been encrypted using the SSL protocol. A padlock icon appears in the browser address bar next to the AMPed URL.

Figure 3: Secure icon

Clicking the icon reveals the site information and the secure certification shown in

Figure 4:

Figure 4: AMPed's Certificate Information

Clicking on the Details tab, the Serial number information can be accessed: **4d 2d a1 bc f7 22 0c 07 0a eb c0 fa 58 82 3f e6**. The serial number is a unique identification number issued by a given certification authority. Having the AMPed site secured means that customer information is encrypted and cannot be intercepted while traveling across the network. A secure site increases trust in a domain and its content. It also boosts ranking in search engines tools.

"It is known that Google prefers sites that are trusted and certified." [7]

Portability

The AMPed graphical user interface (GUI) has a mobile portable design that dynamically adjusts to the proper screen resolution. In order to achieve this cross-browser and cross-devices compatibility, the web interface was developed using the Bootstrap design.

“Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.” [8]

Bootstrap is an open-source JavaScript framework developed by a team at Twitter. It provides a basic structure with Grid System, link styles, and background. It is a combination of HTML, CSS, and JavaScript code designed to help build user interface components which easily and efficiently scale the AMPed website with a single code base, from mobiles to tablets to desktops.

Bootstrap is designed to be responsive to mobile devices. And even though the bootstrap technique was used on the AMPed Web Interface version 1.0, it was not properly implemented. To ensure appropriate rendering and touch zooming, the following **<meta>** tag inside was included into the **<head>** element of all web pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

The **width=device-width** part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The **initial-scale=1** part sets the initial zoom level when the page is first loaded by the browser. [9]

The code displaying the logo in the main page has been enhanced for a proper display on every platform as shown below.

AWI version 1.0 implementation fixed the width and height of the logo, which makes it hard to display on screens with lower resolutions:

```

```

AWI version 2.0 implementation incorporates the bootstrap's ".img-responsive" class.

Responsive images automatically adjust to fit the size of the screen. The width and height attributes do not need to be specified:

```

```

It is important to note that the main core of the Bootstrap framework is the grid layout.

The grid is what dynamically adjusts to the proper screen resolution.

"Grid systems are used for creating page layouts through a series of rows and columns that house your content." [10]

With the Bootstrap's ready-made classes, it is easy to specify how many spots in the grid system each column will occupy. Specifying the number of columns you wish to span, from the available 12, creates the grid columns. In the grid system, one can specify at which point we want the columns to stack horizontally rather than vertically to display properly on any device. The graphic below provides a visual example of how the rows and columns can be stacked:

Figure 5: Bootstrap's grid system

In HTML code, this grid is applied within a `<div>` tag using the 'class' attributes. A `<div>` tag defines a division or a section in a webpage.

In order to work properly, it must define a global 'container' with specific rows and columns. A column could take the full size of the container (ex., `div class="col-sm-12"`) or a portion of it (ex., `div class="col-sm-4"`).

The below example shows the actual code of the AMPed main page; the main title ("AMPed Database") occupies the full width ('col-sm-12') of the central section (`<div>` 'container') within a single 'row', while the 3 main columns ('Peptide database', 'Microbe database', and 'Structure information') each gets an equal portion of that container ('col-sm-4) under another 'row':

```

70 <div class="container">
71 <div class="row">
72 <div class="col-sm-12"> <br>
73 <h1 style="text-align: left;">AMPed Database</h1>
74 <hr>
75 <h3 style="text-align: left;">Browse our content or perform your own <a
href="search.php">search</a>.<br /><br /></h3>
76 </div>
77 </div>
78 <div class="row">
79 <div class="col-sm-4">
80 <h4 style="text-align: center;"><a href="peptide.php"></a><br>
82 <br>
83 Peptide database</h4>
84 <p style="text-align: center;"><a href="peptide.php">Our main focus with AMPed is to
provide all valuable information about antimicrobial peptides in one place. Amino acid sequences
are the most unambiguous way to characterize peptides with diverse naming conventions.</a></p>
85 </div>
86 <div class="col-sm-4">
87 <h4 style="text-align: center;"><a href="microbe.php"></a><br>
89 <br>
90 Microbe database</h4>
91 <p style="text-align: center;"><a href="microbe.php">Peptides that kill microbes are
tested by many different methods, such as in vivo, or in broth, or as inhibition zones on plates,
and antimicrobial activity concentration units must be clearly defined as moles/L or g/L. Test
microbes must be fully characterized by genome and species, which relies heavily on ATCC numbers.
</a></p>
92 </div>
93 <div class="col-sm-4">
94 <h4 style="text-align: center;"><a href="structure.php"></a><br>
96 <br>
97 Structure information</h4>
98 <p style="text-align: center;"><a href="structure.php">Secondary and tertiary structural
information is largely derived from the RCSB database and is classified according to the method
used, such as X-ray, NMR, and Cryo-TEM methods.</a></p>
99 <br>
100 <br>
101 </div>
102 </div>
103 </div>

```

Figure 6: AMPed index page HTML source code

This code running on a desktop device generates the resulting display; the main information is displayed in columns one next to the other; on the top right-hand side there is the header with the 'quick links' section (Search, AMPed Tools, About Us...). The header will be further discussed in more detail, later in this chapter. Below that area is the main search link (search button), next to the AMPed main logo. At the bottom of the page is the copyright information displaying the current year. Just above that is the

footer section with links to the main sections and pages in the web site (this footer section will be discussed in more detail during this chapter).

[Search](#)
[AMPed Tools](#)
[About Us](#)
[Join Us](#)
[Login](#)

Get Started

A powerful search tool with extensive antimicrobial peptides data to support your research needs.

AMPed Database

Browse our content or perform your own [search](#).

Peptide database

Our main focus with AMPed is to provide all valuable information about antimicrobial peptides in one place. Amino acid sequences are the most unambiguous way to characterize peptides with diverse naming conventions.

Microbe database

Peptides that kill microbes are tested by many different methods, such as in vivo, or in broth, or as inhibition zones on plates, and antimicrobial activity concentration units must be clearly defined as moles/L or g/L. Test microbes must be fully characterized by genome and species, which relies heavily on ATCC numbers.

Structure information

Secondary and tertiary structural information is largely derived from the RCSB database and is classified according to the method used, such as X-ray, NMR, and Cryo-TEM methods.

Number of Visits: 5749

SEARCH	AMPED TOOLS	ABOUT US	CONTACT US
Features Tools Tutorial Security	About AMPed Databases Usage Performance	The Team Our Location Research Interests Join Us	How can we help you?

Copyright © 2016 University of Rhode Island. All rights reserved.
 Designed by URI AMPed Team.

Figure 7: Desktop's AMPed index homepage full-width display

The display on a mobile device, on the other hand, despite having all the same information, is displayed quite differently but at the same time, is quite accessible. The below snapshots show the main sections of the site as displayed through a mobile device:

Figure 8: Mobile's AMPed index homepage truncated width display

Branding

As important as the content of a website is, the user or customer experience is a key factor for the overall success of a site. As part of good branding, many factors have to be taken into consideration: an attractive logo, an easy to recognize style, “*an intuitive design*,” [4] and a unique shortcut icon - **favicon**.

The AMPed logo was designed by Eric Zhang, a former URI graduate student and enhanced by Tripti Garg. The logo displays the website identity and allows users to identify with the site’s core brand. It is also a shortcut to the main page; clicking on it takes you back to the AMPed homepage.

Figure 9: AMPed logo

This updated AMPed web interface, AWI version 2.0, follows the original template based on Tripti Garg’s project, AWI version 1.0.

Figure 10: favicon - shortcut icon

“It’s true, favicons are very little things, probably the least important bit of a site, but it’s attention to detail that makes a site stand out.”[11]

The favicon must also be unique. The AMPed favicon was created using as a reference the main logo’s design and adding a transparent background, which can then blend with the browser’s tab and at the same time let the image stand out.

Visitor Counter

A visitor counter or “hit counter” is a computer software program or script that records and displays the number of visitors, or hits, to a particular webpage. Once set up, these counters will automatically be incremented by one every time the web page is accessed in a web browser.

“This counter works in the back ground without providing annoying page delays for the user.” [12]

On AWI version 1.0, the counter was set up so it will increment every time a page is visited regardless of the identity of the visitor: *“A visit counts all visitors, no matter how many times the same visitor may have been to the site.”*[4] For this project, on AWI version 2.0, the script has been upgraded to track only a single visit per session; each distinct visitor’s session has been defined as a 12-hour period. The counter script will

only track the visit once for each visitor regardless of how many times the page is refreshed during a 12-hour session.

The way this is accomplished is as follows: the script hit_number.php tracks a cookie set per user. The counter will not increase if the cookie is still active. If the cookie has expired, the counter will increment by one and record the new number in a text file, 'Countlog.txt'; at the same time, the cookie will be set to start tracking the visitor's new session.

A hit counter allows the Webmaster and users to see how popular the site is. The information provided contributes to make plans for future content. On the other hand, when visitors see how popular one site is compared to others, it encourages them to spend more time exploring the content. This feature is particularly important in a newly formed website like AMPed. The visitor counter is only displayed on the main page.

AWI version 2.0 has incorporated a Google Analytics tool to provide full analytical reporting on all the visited AMPed pages. This free tool provides information to the webmaster on how visitors locate the AMPed site, identifies which pages and links visitors click the most, as well as providing demographics on the users (gender, age range, interests, etc.). It also provides the geographic location of the users using easy to understand graphic reports:

Country ?	Acquisition			Behavior		
	Sessions ? ↓	% New Sessions ?	New Users ?	Bounce Rate ?	Pages / Session ?	Avg. Session Duration ?
	89 % of Total: 100.00% (89)	24.72% Avg for View: 24.72% (0.00%)	22 % of Total: 100.00% (22)	34.83% Avg for View: 34.83% (0.00%)	6.70 Avg for View: 6.70 (0.00%)	00:09:15 Avg for View: 00:09:15 (0.00%)
1. United States	76 (85.39%)	13.16%	10 (45.45%)	38.16%	7.07	00:10:16
2. United Kingdom	10 (11.24%)	100.00%	10 (45.45%)	10.00%	2.80	00:00:00
3. Russia	3 (3.37%)	66.67%	2 (9.09%)	88.89%	10.33	00:14:15

Figure 11: Google Analytics - Geo report of AMPed

One of the most attractive features of using Google Analytics is how easy it can be implemented. The code is based on a few lines of JavaScript code embedded on the footer of all traceable AMPed pages:

```
1 <script>
2   (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
3     (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
4     m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
5   })(window,document,'script','https://www.google-analytics.com/analytics.js','ga');
6
7   ga('create', 'UA-85010664-1', 'auto');
8   ga('send', 'pageview');
9
10 </script>
11
```

Figure 12: Google Analytics code inserted on the footer section of the AMPed's pages

Google Analytics is the most widely used web analytics algorithm. There are many more benefits in this implementation, but getting into more details can be the subject for future discussion and work.

There are pages that do not need to be tracked, like the admin sections of the AMPed site (the User and Peptide admin consoles, which will be explained in more detailed in Chapter 5).

The Header and Footer Sections

The header and footer are key elements in the AMPed site; they not only deliver the “quick links” features, but also provide enough information to encourage users to explore the whole site without getting lost. These two sections have been designed in a

way that a single PHP command is all that is needed to add the sections to all new AMPed webpages:

```
<?php include("footer.php"); ?>
```

As Tripti Garg explained on Chapter 6.2.1 (“Reusable Components”) of her thesis:

“To achieve fast and less error prone development for edits and enhancements, this thesis built AMPed webpages using a component driven design. In order to achieve this, each page, capability and feature was analyzed to check its use and reuse on the site.” [4]

The header and footer provide the unique tone and the style followed by the AMPed site: it starts and ends with a light-green frame matching the background colors of the main page’s banner.

In AWI version 2.0, the header is dynamically designed to reflect whether the user has logged in or not. A new user will see the ‘Login’ link once he/she lands on the homepage:

Figure 13: AMPed header before login

However, once the user has joined the AMPed community and logged in, the header not only reflects the ‘Logout’ shortcut but also personally welcomes the customer:

Figure 14: AMPed header after login

Except for the main page, where the logo is an appreciable part of the main banner, all the other pages include a small reference to the main logo on the top left hand side of the header, pointing back to the main page.

Figure 15: AMPed header with logo reference on intermediate pages

The footer provides access to all the main sections of the site. The AMPed website has adopted the new trend of displaying “mega footers.” This is a larger area where more information is displayed, as opposed to the minimal layout depicted on the header (which has the optimal size and number of links displayed):

SEARCH	AMPED TOOLS	ABOUT US	CONTACT US
Features	About AMPed	The Team	How can we help you?
Tools	Databases	Our Location	
Tutorial	Usage	Research Interests	
Security	Performance	Join Us	
Copyright © 2016 University of Rhode Island. All rights reserved. Designed by URI AMPed Team .			

Figure 16: AMPed footer

In AWI version 1.0, the copyright code in the footer section displayed a static presentation forcing the webmaster to manually update the displayed date at the beginning of each year.

In AWI version 2.0, the copyright code has been automated making the display maintenance-free.

The snippet of code below displays the command used on this enhancement:

```
<?php date_default_timezone_set('US/Eastern'); echo date('Y'); ?>
```

The **date('Y');** command sets the year in a four digit format.

The **date_default_timezone_set('US/Eastern');** sets the default time zone to the US Eastern region.

Figure 17: AMPed copyright info

REFERENCES

- [1] P. DiMaggio, E. Hargittai, W. Russell Neuman, and J. P. Robinson. (2001).
"Social Implications of the Internet." [Online]. Available:
<https://www.princeton.edu/~artspol/workpap/WP17%20-%20DiMaggio,%20Hargittai,%20Neuman,%20Robinson.pdf>
- [2] T. Herrington, JD, PhD, "The Background of Social Issues Affecting Internet Users" in *Linking Russia and America in Web Forums: The Global Classroom Project*. Tech/Web, Eds. Michael Day and Carol Lipson. Erlbaum Publishers, 2005. pp. 167-192.
- [3] G. Konstantinidis, J. Peckham, PhD, L. M. Martin, PhD. (2013). "AMPed - Antimicrobial Peptide Editable Database, Approaches to a Collaborative Design of Bioinformatics Databases." New York Structural Biology Discussion Group 8th Winter Meeting. New York Academy of Sciences, Thursday January 24.
- [4] T. Garg. (2014). "Graphical User Interface for Antimicrobial Peptide Database" (Master's thesis). Retrieved from <http://digitalcommons.uri.edu/cgi/viewcontent.cgi?article=1836&context=theses>
- [5] Hypertext Transfer Protocol -- HTTP/1.1. URL:
<https://www.w3.org/Protocols/rfc2616/rfc2616.html>, Last accessed: 08/30/2016
- [6] What is HTTPS? URL: <https://www.instantssl.com/ssl-certificate-products/https.html>, Last accessed: 09/09/2016

- [7] Z. Ait Bahajji and G. Illyes, "*HTTPS as a ranking signal.*" URL:
<https://webmasters.googleblog.com/2014/08/https-as-ranking-signal.html>,
Last accessed: 09/09/2016
- [8] Bootstrap. URL: <http://getbootstrap.com/>
- [9] Bootstrap Get Started.URL:
http://www.w3schools.com/bootstrap/bootstrap_get_started.asp, Last
accessed: 09/09/2016
- [10] Simply Easy Learning by tutorialspoint.com, "Bootstrap Tutorial". URL:
<http://wiki.lib.sun.ac.za/images/0/07/Bootstrap-tutorial.pdf>
- [11] R. Vellest, "*What's the point of favicons?*" URL:
<http://www.webdesignerdepot.com/2012/11/whats-the-point-of-favicons/>,
Last accessed: 10/01/2016
- [12] What Is A Hit Counter? URL: <http://www.free-hit-counter.com/what-is-a-hit-counter.htm>, Last accessed: 10/01/2016
- [13] K. Noyes, "*Five Reasons Linux Beats Windows for Servers.*" PCWorld. URL:
http://www.pcworld.com/article/204423/why_linux_beats_windows_for_servers.html, Last accessed: 10/01/2016

CHAPTER 3

THE AMPED DATABASE CONTENT

Importing content to AMPed

Initial plans called for the development of an automated bioparser using Perl that would access the National Center for Biotechnology Information (NCBI) database, the largest, most comprehensive collection of genetic and protein information that extracts relevant peptide information and then populate AMPed. The complex public list of NCBI table structures was examined to identify fields that could be used to match and populate our target database table structure. This process was more time-consuming than anticipated.

The information in the NCBI database was found to be inconsistent in format, often duplicated, and in many cases conflicting or irrelevant. Simple categories, such as the name of the peptide, were found in a variety of different fields and in varying formats. A fact to be considered is that NCBI database was developed over 25 years ago, prior to the onset of the big data revolution. Their table structure has not been properly normalized as it dramatically grew over these years; many entries included different forms of the same peptide (pre-, pro-, or individual chains) obtained under different environmental solutions without obvious structural differences (increasing the number of duplications). These ambiguities make it difficult to properly parse synthetic modifications as data was updated. In addition, the integer sequence identifiers (known

as GIs) currently in use by the NCBI are undergoing revisions and may be eliminated in the new future. [1] Changes to the NCBI database format and the basic sequence (FASTA) format for proteins would dramatically affect any bioparser design.

After acknowledging all these issues and with the advice of Professor Martin, a decision was made to switch to the RCSB-Protein Data Bank (PDB) due to the fact that it was smaller, well-curated, and directly based on the molecular structures of the peptides, an important feature and goal of the AMPed project.

The switch to targeting the bioparser to the PDB database was time-consuming as it meant learning and parsing a brand new file format, analyzing which fields were applicable and which ones have non-relevant information, as well as developing a parser that could handle the sheer amount of information available regarding the structure of the proteins. The information from PDB is presented in a standardized text file called the mmCIF Format. The Crystallographic Information File (CIF) data is useful for describing small molecule structures and associated x-ray diffraction experiments.

Proposed matching entries were discovered by students in the CMB/CSC 522 class together with the author by reviewing both raw mmCIF files of PDB entries and through the use of the online PDB to mmCIF dictionary [2]. The dictionary also provided helpful relative usage percentages for many fields within the database. This knowledge was used to eliminate apparently promising PDB fields that were found to be actually used for less than 10% of entries. This arbitrary cutoff was used in choosing fields to source data that would provide AMPed with consistent data, and to avoid generating a large number of AMPed entries with blank fields because there was no PDB data available. A

critical piece of information was absent from most PDB entries: genetic information.

Populating these fields in AMPed by manually accessing the NCBI database and journal articles has become part of the scope for future work to complete the database.

The switch from NCBI to the PDB database as a primary online data source also generated several new fields, such as 'Atom_Type' in the Atomic_Coordinates table (to record a portion of a specific molecule) and other related PDB structures and altering tables that were not a part of the original database structure (DBS) versions 1.0 and 2.0. We found that the basic structure of the database may need to be modified from time to time to accommodate advances in molecular biology and structural biochemistry.

The BioParser

The new BioParser created for this project, known here as BP version 2.0, perpetuates the original mission stated by George Konstantinidis in his thesis, Chapter V: Parsing and Importing Data:

"In order for any loading of the data into the database there must be a process to first find the data files, download them, and then parse them." [3]

As opposed to Konstantinidis' parser, BP version 1.0, where the Go language was used, version 2.0 was developed using Perl (short for "Practical Extraction and Report Language").

"Perl 5 is a highly capable, feature-rich programming language with over 29 years of development." [4]

Perl is an interpreted language; most of its implementations execute instructions directly, without the need to compile a program into machine-language instructions. Perl has become invaluable in bioinformatics because it is good at pattern matching, specifying regular expressions, and performing file and string manipulations. The fact that it has been around for almost 3 decades speaks to its stability. It is easy to read and implement: *“There's more than one way to do it.”* [5] Perl is very portable - it is available for almost every platform. Finally, due to the fact that the Perl language was basically designed with a linguistic rationale, it is more intuitive to a scientist without formal computer science training.

Version 2.0 of the bioparser was developed with the goal to read and import content from a mmCIF format text file downloaded from the PDB database. However, not all files are used in the same way and their information is not represented in a consistent manner. After testing and importing bulk data from PDB into AMPed, it was clear that even this format was not consistent enough in the way the information was stored to perform an automated import of bulk data.

For example, within a mmCIF file there are two different types of tables formatted as follows:

- “List”. Each table is separated by a single line with a single character, the hash or pound sign, '#', known here as ‘the separator’; the separator signals the beginning of the ‘table dictionary’ (any line starting with an underline, ‘_’).

The lines below show an example of a regular .cif file table with the two different columns, one on the left is the field names, the data itself is on the right column:

```
#
_pdbx_struct_assembly.id          1
_pdbx_struct_assembly.details    author_defined_assembly
_pdbx_struct_assembly.oligomeric_details monomeric
_pdbx_struct_assembly.oligomeric_count 1
```

- “Loop”. The ‘table loop’ displaying a list of category declarations followed by a list of data content, like in the example below:

```
#
loop_
_atom_site.group_PDB
_atom_site.type_symbol
_atom_site.label_atom_id
_atom_site.label_comp_id
_atom_site.label_asym_id
_atom_site.label_seq_id
_atom_site.label_alt_id
_atom_site.Cartn_x
_atom_site.Cartn_y
_atom_site.Cartn_z
_atom_site.occupancy
_atom_site.B_iso_or_equiv
_atom_site.footnote_id
_atom_site.auth_seq_id
_atom_site.id
ATOM N N VAL A 11 . 25.369 30.691 11.795 1.00 17.93 . 11 1
ATOM C CA VAL A 11 . 25.970 31.965 12.332 1.00 17.75 . 11 2
ATOM C C VAL A 11 . 25.569 32.010 13.808 1.00 17.83 . 11 3
ATOM O O VAL A 11 . 24.735 31.190 14.167 1.00 17.53 . 11 4
ATOM C CB VAL A 11 . 25.379 33.146 11.540 1.00 17.66 . 11 5
ATOM C CG1 VAL A 11 . 25.584 33.034 10.030 1.00 18.86 . 11 6
ATOM C CG2 VAL A 11 . 23.933 33.309 11.872 1.00 17.12 . 11 7
ATOM N N THR A 12 . 26.095 32.930 14.590 1.00 18.97 4 12 8
ATOM C CA THR A 12 . 25.734 32.995 16.032 1.00 19.80 4 12 9
ATOM C C THR A 12 . 24.695 34.106 16.113 1.00 20.92 4 12 10
ATOM O O THR A 12 . 24.869 35.118 15.421 1.00 21.84 4 12 11
ATOM C CB THR A 12 . 26.911 33.346 17.018 1.00 20.51 4 12 12
# ---- data truncated for brevity ----
```

Even within the same type of table, the content display varies according to the length of the information. For example, the reference entry below could be delineated in two different ways by either semi-colon (;) or by quotes ('), which seems to be chosen based on the content's length.

Example 1 (Entry ID: 1LXE):

```
_citation.title
;Structure of the cathelicidin motif of protegrin-3 precursor: structural insights into
the activation mechanism of an antimicrobial protein.
;
```

Example 2 (Entry ID: 1ZMH):

```
_citation.title      'Reconstruction of the conserved beta-bulge in mammalian
defensins using D-amino acids.'
```

There are many more examples like the above two of how the information from this “standardized” mmCIF format gets displayed in different ways. For this reason, in this project the bioparser was updated to a new version, BP version 3.0, which analyzes PDBML/XML formatted input files rather than the mmCIF format. PDBML/XML Format is based on XML data. XML (Extensible Markup Language) defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

“Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.” [6]

The design goals of XML emphasize simplicity, generality, and usability across the Internet. In this format, the display of information is standardized on every file.

Going back to the two entries on our previous example (Entry IDs: 1LXE and 1ZMH), the revised examples in XML format below shows how they both display their data in a more consistent format making it easier to read and reliable to parse.

Example 1 (Entry ID: 1LXE):

```
<PDBx:citation id="primary">
  ...
  <PDBx:title>Structure of the cathelicidin motif of protegrin-3 precursor: structural
insights into the activation mechanism of an antimicrobial protein.</PDBx:title>
```

Example 2 (Entry ID: 1ZMH):

```
<PDBx:citation id="primary">
  ...
  <PDBx:title>Reconstruction of the conserved beta-bulge in mammalian defensins
using D-amino acids.</PDBx:title>
```

The new XML-based parser, BP version 3.0, using Perl-XML packages can easily and quickly read and parse the downloaded information. Appendix D provides the Perl code that can be used to read a PDBML/XML file.

The only issue that still needs to be addressed with the data downloaded from the PDB site is that it does not provide a way to easily download the needed information in bulk. For that reason, in order for the bioparser to work properly, the information must first be manually downloaded locally to the developer's computer before it is automatically parsed to the AMPed database. An automatic downloader will be a matter of future investigation.

In general, managing and sharing information among structural biology and bioinformatics communities would be a lot easier if there were common standards for representation and storage of peptide and protein data.

Matching Tables: PDB – AMPed

Once the PDB database was selected as our primary content resource, the next step was to match our AMPed tables with their many tables. The AMPed tables, a subset of the diverse types of information available in the PDB, were redesigned to provide information needed to support the microbial peptide research focus of Professor Martin's lab and facilitate collaboration with their colleagues worldwide. The AMPed database currently consists of 17 normalized tables, while the PDB database has 69 tables. The matching process was not a one to one comparison. The information contained within the PDB tables, compared to the ones in AMPed, was scattered over all the 69 tables and fields.

After a thorough analysis, a right set of matches was found. This result can be found in Appendix A.

Entity Relationship (ER) Diagram

To define the AMPed tables in the database we needed to revise the entity relationship (ER) diagram. The ER diagram defines the conceptual view of a database. It shows how

the tables (entities) and their fields (attributes) are defined and how they are related to each other.

The new ER diagram shown in figure 18 illustrates the logical structure of the database.

This is then mapped to the database tables upon implementation; each entity type becomes a table, each single-valued attribute becomes a column, key attributes on the entity types become the primary keys of the table. [7] For a more detailed description of the AMPed ER diagram and a higher resolution graphic, please refer to Appendix E.

**Figure 18: ER diagram of AMPed database
(Courtesy of Sunandha Acharya)**

In this diagram we can see how the entities are related to each other within the AMPed database. For example, at the center of the diagram we have the Peptide entity, with all its attributes (Amino Acid Sequence, Length of Sequence, Molecular Weight, etc.) and

then see how it relates to the Microbe entity by the 'fight against' action. This is the essence of what an antimicrobial peptide is, and forms the central structure of the database. The 3D structure of each Peptide is also represented as an entity in the diagram with its structure amino acid addresses and their respective structures and atomic coordinates branching out. There are 3 types of relationships among entities in the ER, one-to-one, one-to-many, and many-to-many relationships. For instance, the Peptide entity relates to the 3D_structure entity through the 'In the form' relationship. This is a one-to-many relationship, meaning that one 3D_structure instance can be associated with only one Peptides instance; however one Peptide could be associated with many 3D_structure entities. On the other hand, we have a many-to-many relationship between the Peptide entity and the Microbe one. Many different Peptides can fight multiple different Microbes.

The Tables

As previously stated, based on the experience with users of the database and attempts to parse data, the table structures were modified accordingly. In DBS version 3.0, tables like the Atomic_Coordinates, Fight_Against, Inserted_by, Country, and even the User table, were modified; some had new attributes added, others had their types modified. The modifications listed below were added to the current version 3.0 of the AMPed structure.

- The 'Bioparser' and 'Approval_Status' attributes were moved from the 'Peptide' table to the 'Inserted_By' table, this way the 'Peptide' table only has peptide's pertinent information within.
- The 'Resolution' attribute was added to the '3D_Structure' table to reflect the reliability of the experimental data structure in Angstroms. The Angstrom (Å) is an internationally recognized unit of length equal to 1×10^{-10} meters (m) or 0.1 nanometer (nm) commonly used to express the sizes of atoms, molecules, and electromagnetic wavelengths.
- The American Type Culture Collection identification number or 'ATCC_ID' attribute was added to the Fight_Against table. This allows linking the AMPed site to the ATCC website, and eliminates ambiguity about bacterial strains used in testing.
- The relationship between the 'Microbe' and 'Test' tables was renamed to 'Microbe_Test' to be easily identified.
- At the same time the 'Test' table was renamed to 'Test_Results' to put an emphasis on the nature of the data in this table.
- In this table, a new column was added to track the Minimal Lethal Concentration, 'MLC', in addition to the Minimal Inhibitory Concentration since both data are equally useful and informative.
- Under the 'User' table the primary key's attribute, 'UserName', was redefined to match the Email's type. During the process of developing this project we decided to use the user's email address as his/her username due to the fact that each

email address is unique; it makes more sense to have the email address also as the username rather than creating another unique element to use as a username.

The current tables' structure, version 3.0, as approved by Professor Martin along with information about the fields is provided under Appendix B along with a graphical comparison of versions 1.0 and 2.0.

The diagram below shows the current AMPed database with tables, cross-referenced tables, primary keys and constraints applied.

Figure 19: Tables of AMPed database

AMPed Content

One of the main enhancements of the AMPed website is the fact that it can display its valuable content in a clean, clear way. A good example of that is the display of the Peptide table under the Peptide Database page (<https://amped.uri.edu/peptide.php>).

Peptide Database						
Browse our content or perform your own search .						
AMP ID	Accession No	Name	Molecular Weight	Amino Acid Sequence	Notes	Length Sequence
AMP2NLB	P60022	Beta-defensin 1	3891.533	DHYACVSSGGQCLYSACPIFTKIQGTCTYRGKAK CCK		36
AMP2NLC	P60022	Beta-defensin 1	3918.5583	DHYNCVSAGGQCLYSACPIFTKIQGTCTYRGKAK CCK		36
AMP2NLF	P60022	Beta-defensin 1	3950.5141	DHYNCVSSGGQCEYSACPIFTKIQGTCTYRGKAK CCK		36
AMP2NLG	P60022	Beta-defensin 1	3935.4994	DHYNCVSSGGQCLYSACPIFTKIQGTCTYRGKAK CCK		36
AMP2NLH	P60022	Beta-defensin 1	3877.5064	DHYNCVSSGGQCLYSACPIFTKIAGTCYRGKAK CCK		36
AMP2NLP	P60022	Beta-defensin 1	3935.5425	DHYNCVSSGGQCLYSACPIFTKIEGTCTYRGKAK CCK		36
AMP2NLQ	P60022	Beta-defensin 1	3877.4633	DHYNCVSSGGQCLYSACPIFTKIQGTCTYRGAAK CCK		36
AMP2NLS	P60022	Beta-defensin 1	3935.4994	DHYNCVSSGGQCLYSACPIFTKIQGTCTYRGEAK CCK		36
AMP1GQM	P80511	CALGRANULIN C	10443.7319	TKLEEHLLEGIVNIFHQYSVRKGHFDLTKGELK QLTKELANTIKNIKDKAVIDEIFQGLDANQDE QVDFQEFISLVIALKAAAHYHTHKE		91
AMP3PSM	Q68519	Defensin	5500.1712	KTCENLADTFRGPCFTDGSDDHCKNKEHLIKG RCRDFRCWCTRNC		47

Showing 1 to 10 of 41 records.

1 2 3 4 5 Next »

Figure 20: Peptide Database table showing AA Sequences

The same technique is used for browsing the content of the database as well as for displaying the Users' and Peptides admin interfaces (refer to Chapter 5 for more information). The page's design allows the content to be displayed in a dynamic way. To create this page, a technique combining **Asynchronous JavaScript** and **XML (AJAX)** [8] with PHP scripts performs queries to the MySQL database that allows the most efficient

technique for displaying the content without having to reload all the ‘framework’ of the page but focusing exclusively on the desired content.

AJAX reduces the data traffic traveling between the client and the server, which increases response time, performance, and speed. AJAX makes asynchronous calls to a web server. This means that client browsers avoid waiting for all of the data to arrive before rendering of the information to the interface starts. Examples of applications using AJAX include: Gmail, Google Maps, YouTube, and Facebook tabs.

Navigational Features

In order to conveniently browse the AMPed databases and to improve user’s experience, code was developed to parse the retrieved content into sections of 10 rows at the time. A combination of AJAX and PHP code, unique in this thesis, was used to display the total number of displayed records (bottom left on the table) and the current navigational section (bottom right on the table).

Figure 21: Browsing the tables

A set of cascading style commands (CSS) were developed for this project and added to the main css file to handle the look and feel of the navigational code. Under this style, the current selected section gets displayed in a gray square with white font as opposed as the mouse-over feature, which changes the background color to a light blue with a

black font. When an option is not selected, the default style is a white background with a gray font.

This navigational feature makes the site look more attractive and professional; it improves the site's user's experience by helping them to understand where they are and what state they are currently in as they move through the site.

REFERENCES

- [1] NCBI is phasing out sequence GIs - use Accession.Version instead! URL:
<https://www.ncbi.nlm.nih.gov/news/03-02-2016-phase-out-of-GI-numbers/>,
Last accessed: 10/01/2016
- [2] PDBx/mmCIF Dictionary Resources. URL: <http://mmcif.wwpdb.org/>, Last
accessed: 10/01/2016
- [3] G. Konstantinidis. (2014). "*Antimicrobial Peptide Editable Database*" (Master's
thesis). Retrieved from [http://digitalcommons.uri.edu/
cgi/viewcontent.cgi?article=1696&context=theses](http://digitalcommons.uri.edu/cgi/viewcontent.cgi?article=1696&context=theses)
- [4] The Perl Programming Language. URL: <https://www.perl.org/>, Last accessed:
10/01/2016
- [5] S. Cozens, "*Ten Perl Myths.*" URL:
<http://www.perl.com/pub/2000/01/10PerlMyths.html>, Last accessed:
10/01/2016
- [6] Extensible Markup Language (XML). URL: <https://www.w3.org/XML/>, Last
accessed: 10/01/2016
- [7] R. Elmasri and S. B. Navathe, "*Fundamentals of Database Systems*", Seventh
Edition, 2015. Pearson Publishers.
- [8] What is AJAX? URL: https://www.tutorialspoint.com/ajax/what_is_ajax.htm,
Last accessed: 10/01/2016

CHAPTER 4

JOIN US & LOGIN PAGES

Join Us Page

Before this thesis, the only way to add a user to the AMPed database was by direct insertion into the MySQL table by the administrator. This was an inefficient and time consuming process. This thesis provides a web interface that allows users to register themselves to the database through the “Join Us” page. This whole chapter describes the implementation done only during this project.

The "Join Us" page is more than a gateway to our AMPed community; it is also the means by which prospective members are being introduced to the community. It is the first opportunity to build trust. For that matter, one of the main goals of this project was to make the intro page and the ‘Join Us’ form easy to find and comprehensive. There are links to the "Joins Us" page on both the header and footer of the website.

Figure 22: "Join Us" link at header section

To make the application process run smoothly, we made sure that all of the information needed to complete the form is accessible; nothing detracts more from the goal of a

welcoming form than making a prospective member go look for information needed to complete the form.

The image shows a web form titled "Join Us". Below the title is a horizontal line, followed by the instruction "Please fill out all fields." and another horizontal line. The form contains nine input fields, each with a label to its left and a placeholder text inside the input box. The fields are: "Email address (also used as username)" with placeholder "Please enter your Email address."; "First Name" with placeholder "Please enter your First Name."; "Last Name" with placeholder "Please enter your Last Name."; "Job Title" with placeholder "Please enter your Job Title."; "Affiliation" with placeholder "Please enter your Affiliation."; "Telephone" with placeholder "Please enter your Telephone."; "Address" with placeholder "Please enter your Address."; "Building-Mail/Stop" with placeholder "Please enter your Building-Mail/Stop."; and "City" with placeholder "Please enter your City.".

Field Label	Placeholder Text
Email address (also used as username)	Please enter your Email address.
First Name	Please enter your First Name.
Last Name	Please enter your Last Name.
Job Title	Please enter your Job Title.
Affiliation	Please enter your Affiliation.
Telephone	Please enter your Telephone.
Address	Please enter your Address.
Building-Mail/Stop	Please enter your Building-Mail/Stop.
City	Please enter your City.

Figure 23: Join Us page fields

Since the user's email address also serves as his/her username, it is the first field to be input. It is also the only field to be 'live authenticated'; using a combination of AJAX and PHP commands, an instant query is placed to the 'User' database to verify that the customer is not already registered. If the user's email address is not in the database, the user can proceed to fill out all the other required information. Except for the "Building-Mail/Stop" field, the user is required to fill in all of the other information in order to register. If the user's email already exists in the database, the customer is notified and the page is redirected to the login page. If the customer does not remember the password or cannot find the received email with the login credentials, he/she must

contact the administrator and request his/her password be reset. The password is the only element of the form that gets encrypted prior to being added to the database.

Before submitting the form, the customer has to make sure that the information provided in the “Enter Image Text” field matches the CAPTCHA image. More information about this feature will be provided further down in this chapter.

Validation Process

Ideally, customers will fill in any web forms with required information and submit their data entry successfully. However, people often make mistakes. This is where web form validation plays an important role in the tool design. The first and most obvious data that should be validated is the required information – information without which an operation cannot be completed successfully. Thus, validation ensures that the user provides all the necessary details to the web form and the process fails if at least one of the fields is not filled.

A user's input can be validated either on the server side or on the client side (web browser) or both. In server-side validation, information is sent to the server and validated using the server-side language (PHP code, for this project). If the validation fails, the response is then sent back to the client's browser, the form is refreshed, and feedback is shown. This method is secure because it will work even if JavaScript is turned off in the client's browser and malicious users cannot easily bypass it. The downside is that users fill in the information without getting a response until after

submitting the form. This results in a slower response from the server and an annoying experience for the user even if this is the safest way to make sure all the required information has been provided.

A client-side validation process is done on the client's web browser by employing script languages such as JavaScript. With client-side validation, the form data never gets submitted if the validation fails. By the script language, a user's input can be validated simultaneously as they type (in-line form validation). This means the process is more responsive, providing a visually rich validation of their effort. Validation is being handled by JavaScript methods and the users get immediate feedback if their data validation fails.

The main drawback of client-side validation is that it relies on interpretative languages such as JavaScript. If users turn their host JavaScript interpreter off, they can easily bypass the validation.

A third way to validate a form, also within the client-side validation process and a complement of the previous method, is through the *“required”* attribute. The *“required”* attribute is a boolean feature which *“specifies that an input field must be filled out before submitting the form.”*[1]

The AMPed web forms implement a triple validation process. For example, in the Contact Us form, this triple validation process has been implemented for all fields; here is a closer look at the First Name field validation in JavaScript, HTML code, and PHP script.

JavaScript validation code:

```
if (!isFilled(form.First_Name.value)) { //Checks if the field is not null
    errorMessage += "\tYour First Name.\n";
    displayMessage = true; // displays error message as a pop-up window
}
```

HTML code using the “required” attribute:

```
<input style="height: 35px; width: 100%;" placeholder="Please enter your First
Name." name="First_Name" id="First_Name" required type="text">
```

PHP code validating all required fields:

```
if (!$First_Name || !$Last_Name || !$Email || !$comments) {
    $_SESSION['incomplete'] = "<p id='incomplete'>Please fill out all the fields before
submitting the form. Thank you.</p>";
    header ("location: contactus.php"); exit();
}
```

By combining server-side and two client-side validation methods, we get the best of both: fast user response, more secure data validation, and overall a better user experience.

Once a user’s required fields have been validated, experimental data is added to the appropriate tables in the AMPed database together with a ‘pending’ (revision) status.

Registration Process

As soon as the customer registers, the site generates an eight-digit random password based on a combination of: upper and lower case characters, the ten single-digit numbers, and the hyphen (abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ23456789-).

The password gets encrypted and along with all of the other information it gets validated and added to the User table. More information about the validation process will follow.

Once the information has been entered into the User table, an email is sent to the registrant acknowledging the registration process and providing his/her credentials. The snapshot below displays an example of the email:

Figure 24: Welcome to AMPed registration email

A similar email goes to the AMPed administrator informing them about the potential new community member. The administrator, through the “User Portal” interface has access to all the information submitted by the User and can update the User’s access level along with any other information.

By default any new registrant starts with level 4 access. More information about the access levels will follow.

Login Page

As with any major heterogeneous database, the AMPed database is an open research tool; a user does not need to be registered to browse, search, or download useful information. But in order to be a functioning part of the AMPed community one must register and log into the website.

The login page works as a threshold for many of the aspects or tools offered in AMPed.

As described in Tripti Garg's thesis, three fields are required to submit the login form:

1. *"User Name: referred to as an account name, is a string (i.e., sequence of characters) that uniquely identifies a user. User name in AMPed is a completely arbitrary value assigned by the AMPed administrator."*
2. *"Password: similar to User Name, password is a string, but it differs from a user name in that it is intended to be known only to its user. Passwords do not display in clear text in the AMPed GUI."*
3. *"CAPTCHA: is a graphic presented with distorted text used to tell whether the user is a human or a computer."*[2]

Search Join Us Contribute About Us Login

AMPed
Antimicrobial Peptide Database

Login to your account

User Name

Password

qh68

Enter Image Text (case sensitive)

Don't have an account? [Join Us](#)

Need help? [Contact Us](#)

Figure 25: Login page

“When the user attempts to log into the system, a Captcha image is first generated by the system and compared to the text entered by the user. If correct, the user names and passwords entered by the user are compared with data contained in special User tables in the AMPed database. If successful, user is provided access to the AMPed site.” [2]

As implemented in AWI version 2.0, if the customer does not have an account, the login page now provides a new option to join the community and register; a link pointing to the ‘Join Us’ page has been added at the bottom of the form. There is also an option to contact the administrator, through a link to the ‘Contact Us’ page; in case the customer needs additional assistance.

Another feature added in AWI version 2.0 is that once successfully logged in, the system will welcome the user into the community. A welcome message with the user's first name gets displayed right below the header section:

Figure 26: Welcome message header

The message will appear on every page until the user ends the session (logs out from the site).

Access levels

On the previous AMPed web interface version, there was no access level implementation. This is a new enhancement unique to this AWI version 2.0.

In this implementation there are 5 access or permission levels set in the AMPed backend interface. Here they are explained in order from lower to a higher degree of access.

- Level 5. This is the lowest access level. Any user browsing or using the search tools, before being logged in, is by default a level 5 user - an anonymous user.
- Level 4. Once the user has joined the community (successfully filled out the 'Join Us' form) and has logged into the website, they are by default granted this access level. At this moment his/her privileges in the AMPed site are not different from those of an anonymous user except that he/she gets a more

personalized page (welcome message at the top right corner of every page and automatic self-populated contact info when using the 'Contact Us' form). Also the user's IP address gets registered in the site's log file. Access levels in the AMPed site can be changed; it is left to the discretion of the administrator or moderator to change a user's level up or down.

- Level 3. This level is reserved for researchers or faculty members studying Antimicrobial Peptides. At this level the user can insert new entries and upload peptide information to the database - the 'contribute' page becomes available.
- Level 2. This level is reserved for software developers or graduate students working directly with or under Professor Martin's supervision. They are responsible for maintaining and troubleshooting any issue with the AMPed backend.
- Level 1. This is the highest access level assigned by the AMPed site: the administrator level. This level provides access to modify any of the user's information (from changing access level, to resetting passwords, as well as modifying any information submitted). At the same time, the administrator can approve or reject any contribution made to the AMPed community. Refer to Chapter 5 for more information.

CAPTCHA

Any web administrator or website owner is aware of the unpleasant task of dealing with spambots. A spambot is a computer program designed to collect, or harvest, e-mail

addresses from the Internet in order to build mailing lists for sending unsolicited e-mail, also known as spam. A spambot can gather e-mail addresses from Web sites, newsgroups, special-interest group (SIG) postings, and chat-room conversations. [3] Spambot activity can lead to everything from harassment of other users (spamming them with unwanted links or information), to creation of links to bad websites, or just being annoying in general.

The most efficient way to fight against these spambot web crawlers is with CAPTCHAs. The word CAPTCHA is an acronym developed by computer scientists at Carnegie Mellon University in 2000 meaning the following:

Completely **A**utomated **P**ublic **T**est to tell **C**omputers and **H**umans **A**part.

In other words, these tools work as gatekeepers controlling your entry into the database by judging whether you are a human or a spamming machine.

Figure 27: AMPed CAPTCHA image

Computer scientists have found out that the best way to keep away spammers is to use language images. A randomly generated text is generated and the image is manipulated, so that a human can just read the same while a computer trying to capture a the text cannot.

Within the AMPed site, CAPTCHAs have been placed on the 'Login', 'Join Us', and 'Contact Us' pages.

This project AWI version 2.0 has built up and improved the CAPTCHA feature originally designed AWI version 1.0 but maintained its functionality:

“To create CAPTCHA for AMPed, this thesis used PHP’s GD library image functions [24][6]. The GD Graphics Library is a graphics software library for dynamically manipulating images. GD can create images composed of lines, arcs, text, other images, and multiple colors. Its native programming language is ANSI C... , for AMPed CAPTCHA, we chose and built the image in JPEG format.” [2]

To make it easier for humans to read, the CAPTCHA code has been configured to avoid the possible situation in which the image could be ambiguous. In that regard, the letters and numbers below will not appear on the image: upper case vowel “O”, upper case vowel “I”, and lower case consonant “l”, as well as the numbers: ‘0’ and ‘1’.

For this project, AWI version 2.0, the 4-character code has perfectly aligned in the center and middle within the CAPTCHA image to make it easier for users to read as opposed to the previous version 1.0 where the text appeared on the top left portion of the image generated making it harder to read and more distracting.

REFERENCES

- [1] HTML <input> required Attribute. URL:
http://www.w3schools.com/tags/att_input_required.asp, Last accessed:
10/01/2016
- [2] T. Garg. (2014). "*Graphical User Interface for Antimicrobial Peptide Database*"
(Master's thesis). Retrieved from [http://digitalcommons.uri.edu/
cgi/viewcontent.cgi?article=1836&context=theses](http://digitalcommons.uri.edu/cgi/viewcontent.cgi?article=1836&context=theses)
- [3] Antispam: Protect Your Website Against Spammers. URL:
<http://antispam.byteplant.com/products/cleanmail/spambot-trap.html>, Last
accessed: 10/05/2016
- [4] CAPTCHA: Telling Humans and Computers Apart Automatically. URL:
<http://www.captcha.net/>, Last accessed: 10/05/2016

CHAPTER 5

ADMINISTRATOR SITE

In version 1.0 of the AWI, there was not an Administrator site for controlling the function of the AMPed database. This whole chapter describes the concept and implementation of this tool performed during this thesis. Like every website there is a need for a webmaster, moderator, or administrator who oversees and safeguards the site and its content. This website 'superintendent' has complete access to the domain and its content. The Administrator has a set of tools tailored to his/her needs and only accessible through the main login page. Furthermore, none of the admin tools can be accessed directly (even with the URL bookmarked) unless the user has properly logged into the website and has the appropriate access level.

The snippet of code below displays this secure feature:

```
1 <?php
2 session_start();
3 if ($_SESSION['is_auth'] != true) {
4     $_SESSION['redirect'] = "_admin/index.php";
5     header ("location: ../login.php"); exit();
6 }
7 // Determines than only levels 1-2 can access content
8 if (isset($_SESSION['Level'])) {
9     if ($_SESSION['Level'] > 2) {
10         header ("location: ../amped_tools.php"); exit();
11     }
12 } else { header ("location: ../amped_tools.php"); exit(); }
13 ?>
```

Figure 28: Secure access to AMPed Admin Tools

Line 2: sets and starts a new logging session (period of use) or resumes an existing session.

Line 3: checks whether the logging session has been authenticated.

Line 4: If not already set, it creates a session variable ('redirect') to hold the location of the current page. This will allow the site to redirect to this page after a successful login.

Line 5: redirects the page to the login form.

Line 8: if the login session has been authenticated it checks whether the user's access level has been set.

Line 9: if so, it checks whether the user's access level is 2 or 1 (only access level 1 and 2 are granted access to the admin page).

Line 10: if the user does not have the proper credentials, the page will redirect to the 'Search' page.

Line 12: if the user's access level has not been set it will redirect the page to the 'AMPed Tools' page.

The above code makes sure that the page is only accessed by the administrator after a successful log in.

User Portal

The User Portal is the interface where the administrator can validate the user's credentials, change the user's access level, or reset his/her password. By default the listing is sorted in ascending order by the user's first name. At the bottom of the table it shows the total number of records on the left hand side as well as the navigational option on the bottom right corner. These are the same convenience features explained in Chapter 3 for displaying search results.

The 'User Portal' has been designed so it will display all content on a single table with one column per field, as shown in figure 29.

User Portal									
UserName	Access Level	First Name	Last Name	Email	Contact No	Job Title	Affiliation	Country ID	Address
abraham_herrera@my.uri.edu	2	AbrAhAm	HerrerA	abraham_herrera@my.uri.edu	4015365560	Grad student	SIMULIA	US	108 HIGH SERVICE AVE
ashleyb91495@my.uri.edu	4	Ashley	Bottella	ashleyb91495@my.uri.edu	4014739952	Student Researcher	University of Rhode Island	US	77 Silver Spring Ave
ekaterina.grafskaya@phystech.edu	4	Ekaterina	Grafskaya	ekaterina.grafskaya@phystech.edu	+89645756425	PhD student	MIPT	RU	Institutskiy per. 9
user2	3	FName2	LName2					AF	
imdextradeur@my.uri.edu	4	Ian	Dextradeur	imdextradeur@my.uri.edu	401-598-6408	Student	University of Rhode Island	US	26 Walker st. apt G
jboyntonjr2@gmail.com	4	Jack	Boynton	jboyntonjr2@gmail.com	4018852421	Research Student	University of Rhode Island	US	30 Benjamin St.
Steadhead@mac.com	4	Joseph	Steadman	Steadhead@mac.com	4018354185	Student Researcher	University of Rhode Island	US	1279 South Road
joshua.gyllinsky@gmail.com	2	Joshua	Gyllinsky	joshua.gyllinsky@gmail.com	000*	Developer	URI	US	
Martin@uri.edu	1	Lenore	Martin	Martin@uri.edu	401.874.5049	Associate Professor	University of Rhode Island	US	9 East Alumni Avenue

Figure 29: AMPed - User Portal snapshot

The administrator can access and modify any entry by clicking on the UserName column. This will pop up a new window, which will allow the administrator to modify/update the user's entry.

Fetched data from Username: user2@email.com	
User Name:	user2@email.com
Password:	xxxxxxx
Access Level:	3
First Name:	AbrAhAm
Last Name:	HerrerA
Email:	user2@email.com
Contact No:	4015315000
Job Title:	
Affiliation:	URI
Country:	United States
Address:	1301 Atwood Ave
Building:	
City:	Johnston
State:	Rhode Island
Postal Code:	02919
Note:	Field for the Administrator to add notes or comments.
<input type="button" value="Submit"/>	

Figure 30: User's Portal pop-up window

Other than the UserName, all other fields can be modified. Once selected, the username is the only field that will remain the same for a specific user. As indicated before, the username and the email share the same information; the project wanted to minimize the user's ability to create their own username and password. Since the email address is unique to any user, it was decided that it will also be used as the username. Within this pop-up window there is a 'Note' field where the administrator can record any pertinent comments or notes regarding the user. Those comments cannot be

accessed or seen by the customer. Once the entry has been updated and the pop-up window is closed, the User Portal interface gets automatically updated to reflect the new information or status.

Peptide Portal

As with the User portal explained in the previous section, the Peptide portal is unique to AWI version 2.0. When a new entry gets added to the peptide database, by design it goes into a “Pending” approval state. Peptide entries with this status will not appear on the public listing when browsing the Peptide database page, nor will they get displayed through the search result pages. Only the administrator can review and approve any entry for public release.

The Peptide Portal follows the same design as the User Portal - the information is displayed in a table with columns marking each field. Below the table the total number of peptides in pending review mode is displayed on the left hand side as well as the navigational option on the bottom right corner.

The Administrator has the power to approve, reject, or leave in pending mode any uploaded data or contribution. At the same time, the reviewer can modify (update) any of their submitted fields in the AMPed database except for the AMP_ID (designated by the AMPed tool) and the contributor’s information (those fields are displayed in the pop-up window with a grey background).

Fetched data from Peptide: AMP1L34	
AMP_ID:	AMP1L34
Status:	Pending ▼
Peptide:	Beta-defensin 1
AA Sequence:	DHYACVSSGGQCLYSACPIFTKIQTGYRGKAKCCK
Species Name:	
ATCC #:	
Test method:	X-RAY DIFFRACTION ▼
MIC:	15
UOM:	mg/L
Inserted By:	
First Name:	Ekaterina
Last Name:	Grafskaya
User Name:	ekaterina.grafskaya@phystech.edu
Note:	This field is for the Administrator to enter comments.
<input type="button" value="Submit"/>	

Figure 31: Peptide contribution by user

Through the Peptide console, the data reviewer is provided with email communication with the data contributor in case more information is needed before approving the contribution and making it publicly available for anyone to access.

CHAPTER 6

SEARCH SECTION

Search Page

The “Search” page implemented on this project has respected the original goal and design philosophy set out by AWI version 1.0:

“The AMPed search needed to be flexible to meet the needs of the range from expert to novice users and be able to process a variety of criteria through tens of thousands of records in the AMPed database tables. The search had to be designed for speed and accuracy. The search also had to manage a variety of data types.” [1]

To be an effective page, a clean and concise design of the search page was implemented. Only a few of the most relevant search keys were allowed into the “Search” page; these selected fields are:

- Peptide Name
- Amino Acid Sequence
- ATCC Number
- Species Name
- AMP ID, and
- PDB ID.

This thesis has enhanced the overall user experience by adding a *'placeholder'* attribute to each input field. The *'placeholder'* attribute specifies a short hint that describes the expected value of an input field. [2] The *'hint'* disappears as soon as the user starts typing in content. For example, “Enter a peptide name...” is the placeholder attribute for the Peptide Name field.

Figure 32 displays the current state of the page as implemented in the AWI version 2.0:

The screenshot displays the AMPed Search page. At the top, there is a green header bar. Below it, the AMPed logo (Antimicrobial Peptide Database) is on the left, and navigation links (Search, AMPed Tools, About Us, Join Us, Login) are on the right. The main heading is "Select Search Criteria". Below this, a prompt says "Enter at least one search criteria.". There are six search criteria fields, each with a label and an input box containing a placeholder text:

- Peptide Name: Enter a peptide name...
- Amino Acid Sequence: Enter an amino acid sequence...
☐ Full Match
- ATCC Number: Enter an ATCC Number...
- Species Name: Enter a Species Name...
- AMP ID: Enter the AMP ID...
- PDB ID: Enter the PDB ID...

A "Search" button is located at the bottom of the form.

Figure 32: AMPed Search page

This project has also enhanced the AWI version 1.0 by adding a “live search” feature on all the pages and forms, where a query to the database is required. By applying an AJAX-PHP combination, the goal is to see possible search results without needing to refresh the page or redirecting to a results page.

The AJAX code has been customized to work with all the major browsers (Firefox, Chrome, Opera, Safari, and Internet Explorer).

The code snippet below shows the AJAX code setting up an event listener to capture the user’s keystrokes. Based on this input, the captured information gets sent to a PHP script, which at the same time performs a real-time query to the database and submits the result back to the AJAX code, for immediate display on the current page.

```
1 <script>
2 function showPeptide(str)
3 {
4     if (str.length==0)
5     {
6         document.getElementById("peptide-1st").innerHTML="";
7         document.getElementById("peptide-1st").style.border="0px";
8         return;
9     }
10    document.getElementById('peptide-1st').style.display='block';
11    if (window.XMLHttpRequest)
12    { // code for IE7+, Firefox, Chrome, Opera, Safari
13        xmlhttp=new XMLHttpRequest();
14    }
15    else
16    {
17        // code for IE6, IE5
18        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
19    }
20    xmlhttp.onreadystatechange=function()
21    {
22        document.getElementById("peptide-1st").innerHTML=xmlhttp.responseText;
23        document.getElementById("peptide-1st").style.border="1px solid #A5ACB2";
24    }
25    xmlhttp.open("GET","get-peptides.php?searchstr="+str,true);
26    xmlhttp.send();
27 }
28 </script>
```

Figure 33: AJAX code to display Peptides results as the entry is input

When the information the user is looking for is present in the database, a list of possible options appears right below the input text, as shown in the screenshot below. The user can easily click to select any of the possible matches listed. To make it easier to identify the nature of the hit, the resulting list highlights all the entries matching the original input string in a red font.

The screenshot shows a search interface with a table-like structure. The first column contains labels for different fields: 'Peptide Name', 'Amino Acid Sequence', 'ATCC Number', and 'Species Name'. The second column contains a search input field and a dropdown list of results. The input field contains the text 'ne'. The dropdown list displays five entries, each with the word 'neutrophil' in red to indicate a match with the input. The entries are: 'Human neutrophil peptide 1', 'neutrophil defensin 1', 'neutrophil defensin 1 (HNP-1) (HP-1) (HP1) (Defensin, alpha 1)', 'neutrophil defensin 2', and 'neutrophil defensin 4'.

Field	Value
Peptide Name	ne
Amino Acid Sequence	Human ne utrophil peptide 1
ATCC Number	ne utrophil defensin 1
Species Name	ne utrophil defensin 1 (HNP-1) (HP-1) (HP1) (Defensin, alpha 1)
	ne utrophil defensin 2
	ne utrophil defensin 4

Figure 34: Live search with a list of possible matching results

An additional feature has been added to the 'Amino Acid Sequence' input field, in that any character typed in automatically converts to uppercase; that is how an Amino Acid Sequence is traditionally displayed, saving the user the trouble of holding down the Shift key.

By design, the match list has a maximum of 7 'unique' rows displayed per hit; if there is more than one exact match, the list will only display one occurrence.

If none of the characters on the search string matches an entry in the database, a 'no suggestion' message is displayed instead of the list option. The 'no suggestion' message cannot be clicked on (selected) like it can when a matching hit is encountered.

Peptide Name	Petit
Amino Acid Sequence	no suggestion
	☐ Full Match

Figure 35: Live search without a matching hit

Due to the biological nature of an amino acid sequence string, the search page offers two options for searching: by default the search engine provides a ‘partial’ match search but one can also choose to only perform a ‘full match.’

“The full and partial search option on AA sequences allows researchers to quickly and accurately find results that might be of interest to them. The example below highlights how a full and partial string search using an amino acid sequence provides different results when searched via AMPed.” [1]

MNAKYDTDQGVGRMLFLGTIGLAVVVGGLMAYGYYYDGKTPSSGTSFHTASPSFSSRYRY
1 search result found

MNAKYDTDQ**GVGRMLFLG**TIGLAVVVGGLMAYGYYYDGKTPSSGTSFHTASPSFSSRYRY
MSLYLLGLKILRYLKMV**IGVGRMLFLG**AFLLSVKFLREKRRLKMYLGIMLGF
MVRRLAVRANMPKDSLARDSLPKDSLARDFLSDKTSPTDGTQSG**GVGRMLFLG**TAVDYVHLT
MNIPKTCFQIHNKIQ**GVGRMLFLG**YLIRINLNIFLIYHFSPYCPYLFTVFNSLINLI
MDNEIIIVIIIIFFYLKQKKETQVVKVQK**GVGRMLFLG**NLKLKLNK
5 search results found

Full Match
Searches the entire database for exact matches for the amino acid sequence that the user provides and displays the results.

Partial Match
Searches the entire database for all amino acid sequences that contain the sequence that the user provides and displays the result.

Figure 36: Full vs. Partial match
(Image courtesy of Tripti Garg)

Search Criteria

The search page allows a user to search for specific information in the AMPed database.

Once the form is submitted, a list of all the entries matching the criteria will be displayed along with the number of results found.

The information will be displayed in a table with the four main columns: AMP ID, Accession Number, Peptide Name, and the Amino Acid Sequence.

The AMP ID column is a link to a page with more detailed information about each peptide. In this page, along with the basic information, the user can see information about the Microbes and Species related to the Peptide, its structure, and if available, the Journal (with Article's Title, Author(s), year of publication, etc.) in which additional information appears.

The Accession Number column is linked directly to the relevant entries in the UniProt database. The Universal Protein Resource (UniProt) is a comprehensive resource for protein sequence and annotation data maintained by the European Molecular Biology Consortium. [3]

Advanced Search

The 'Advanced Search' page allows a user to search for specific types of information in the AMPed database. A user can pick and choose one or more types of data to search for throughout the entire database.

Select Search Criteria

Basic Information

☐ Peptide Name
☐ AA Sequence
☐ ATCC Number
☐ Species Name
☐ AMP ID

Microbe Information

☐ Microbe Type

3D Structure

☐ AA Name
☐ Atom Name
☐ XYZ Coordinates
☐ PDB ID

Genome Information

☒ DNA Sequence
☐ Genome ID
☐ Species
☐ Chromosome Location
☐ RNA Transcript

DNA Sequence

Figure 37: Advance Search page

The advance as well as the basic search features are based on fulfilling two main goals: quality and speed. With the growing demand for peptide research, and the need for fast and accurate results, the AMPed website provides the needed tools for this area of research to succeed and grow.

These tools are highly customizable and can adapt quickly to any future requirements.

REFERENCES

- [1] T. Garg. (2014). *"Graphical User Interface for Antimicrobial Peptide Database"*
(Master's thesis). Retrieved from [http://digitalcommons.uri.edu/
cgi/viewcontent.cgi?article=1836&context=theses](http://digitalcommons.uri.edu/cgi/viewcontent.cgi?article=1836&context=theses)
- [2] HTML <input> placeholder Attribute. URL:
http://www.w3schools.com/tags/att_input_placeholder.asp, Last accessed:
10/10/2016
- [3] UniProt. URL: <http://www.uniprot.org/>, Last accessed: 10/10/2016

Chapter 7

ABOUT US

The 'About Us' page provides an introduction to the team, the AMPed project, as well as the team's location. It also provides a link to the join the community through the 'Join Us' page. In the AWI version 1.0 it only provided a placeholder for the information. The current AWI version, 2.0, has provided actual and meaningful content.

Figure 38: About Us page

For future enhancements, the 'About Us' should include the mission and vision of the AMPed site.

CHAPTER 8

CONTACT US

Many websites neglect the importance of having contact information in an easy to find place. Some others do not even bother to have one. The 'Contact Us' page is in fact one of the top four most important pages on a website. [1] With our primary goal of creating an online community of antimicrobial researchers from around the world, this is an essential part of opening a dialog between our university and the world.

This project has designed and implemented a fully functional 'Contact Us' page, not present on the previous AWI version, where users can submit questions, comments, or suggestions. The content of this page clearly states its purpose; it provides different methods of contacting the site owners, and it has a defined 'call to action.' In the AMPed site, the Contact Us page is linked from the footer section of all web pages.

If an anonymous user is accessing the page, the form will start with the required fields empty (as shown in figure 39). On the other hand, if the user has already registered and logged into the community, the form will be auto-populated with his/her first name, last name, and email, making it easier for the user to submit multiple times.

In order to prevent spam, the CAPTCHA field has been incorporated within the form.

All fields are required in order to submit the form.

Figure 39: Contact Us page

Once the form is submitted, the administrator will receive an email with all the pertinent information. The administrator will be able to reply back using the customized and newly created AMPed email: amped@uri.edu. This customized email was created as a direct request from this project's main developer, to keep the lines of communication with the AMPed community uncluttered.

The Contact Us page increases the sense of community on the AMPed site.

REFERENCES

- [1] Neil Patel, "*The 4 Most Important Pages on Your Website (& How to Optimize Them).*" URL: <http://blog.hubspot.com/marketing/optimize-important-website-pages>, Last accessed: 10/10/2016

Chapter 9

CONCLUSION AND FUTURE WORK

The Anti-Microbial Peptide Editable Database (AMPed) is a high quality annotated collection of data describing antimicrobial peptides, their sources, and their targets. It has been in development for several years under the direction of Professor Lenore Martin as part of the project oriented graduate class, CMB/CSS 522 Bioinformatics, at URI. Its creation has been the work of many dedicated undergraduate and graduate students, as well as guided by many Professors and graduate advisors.

With Professor Martin's guidance, all the goals set at the beginning and during this project to date have been achieved. The primary goal was to globally improve AMPed, with a focus on supporting the ability for the research community to insert and Professor Martin to curate the data. However, in order to complete the interface, several other system and database design issues had to be resolved. The below listing summarizes the achieved goals.

- Transferred the AMPed code to a secure new virtual server with higher capacity and faster performance.
- Established the AMPed's own domain, amped.uri.edu, and its own administrator email account, amped@uri.edu.
- Enhanced the PHP code to run under a new secure protocol server and to make it portable; easy to access and run from any web device.
- Populated the AMPed site with a new Bioparser capable of reading and manipulating CIF and XML files from the PDB site.

- Enhanced visitor counter to track unique visitors rather than page loads.
- Customized header with user level's access and greeting message.
- Enhanced the footer section with google analytics tool.
- Automated copyright year display.
- Enhanced user experience by adding live search capability and all needed pages.
- Linked information to other databases , such as UniProtKB and PDB sites
- Implemented the Join Us page which adds a user to the database and communicates it to the administrator
- Developed an access level implementation
- Enhanced the AMPed community by adding a contribution tool to manually insert peptide information
- Provided a Peptides Portal interface to monitor peptide contributions.

In short, the AMPed database and website has all the user and administrative features to become the number one repository of antimicrobial peptide information worldwide. Appendix C shows a comparison of AMPed features with the other antimicrobial peptide repositories currently found online.

In spite of all these accomplishments, the work in the AMPed site is not done; many new features could be and should be added to the AMPed website. As previously mentioned in Chapter 7, the Mission and Vision of the site should be available under the 'About Us' page.

Another important enhancement to consider is a 'Privacy Policy'. Since this community is collecting personal information, it is a good idea to assure the customers that their private information will not be shared with any third party company without their written permission. In addition to this, a 'Terms of Use' ('Terms and Conditions') should be added to the site. A website terms and conditions page may be used to satisfy legal disclosure obligations, such as granting users rights to use website materials, impose acceptable use obligations, to limit (or attempt to limit) warranties and disclaim liabilities, and more generally to structure the legal relationships between the website operator and users. [1]

For this project, all the AMPed webpages are being tracked with Google Analytics code. In order to get the maximum benefits of this feature, further work needs to be done to analyze and report on visitor's activities, check what pages or sections are more popular, and enhance the pages needing more visibility.

Finally, as has already been stated, a new tool to automatically download content from a variety of sources and parse it automatically to the AMPed site might need to be developed, to continue populating the site with the ever-expanding amount of new, validated content.

REFERENCES

- [1] Free Website Terms and Conditions. URL: <http://www.contractology.com/free-website-terms-and-conditions.html>, Last accessed: 10/10/2016

GLOSSARY

AMPed

The Antimicrobial Peptide Editable Database developed at the University of Rhode Island under the direction of Professor Lenore Martin.

Antimicrobial

An antimicrobial is an agent that kills microorganisms or inhibits their growth, which includes antibacterial, antiviral, antifungal and antiprotozoal agents.

AJAX

AJAX stands for Asynchronous JavaScript and XML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script. Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.

Amino acid

Organic compound that serves as the building blocks of proteins.

AWI

AMPed Web Interface; the group of web pages and forms that access the AMPed database.

Bioparser

Script designed to download information from PDB database, annotate, and then populate the AMPed database.

Bootstrap

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language.

DBS

Database Structure; the structure of the tables conforming the AMPed database.

Dehydration synthesis

Dehydration synthesis is the reaction that occurs when two amino acids are initially bound together during protein synthesis. The reaction results in the release of water, a hydrogen from the amino end of the new amino acid and an hydroxide ion from the carboxyl end of the existing protein.

FASTA

A text-based format for representing either nucleotide sequences or peptide sequences, in which nucleotides or amino acids are represented using single-letter codes.

Gestalt Principle

Gestalt principle or gestalt laws are rules of the organization of perceptual scenes. It describes how people organize visual elements into groups or unified wholes. Gestalt is also known as the "Law of Simplicity".

MIC

Minimum inhibitory concentration (MIC) is the lowest concentration of the peptide that will inhibit the growth of a targeted microorganism.

Molecular weight

Molecular weight is the total mass of a given peptide as determined by the sum of the composite amino acids (as determined by NIST) minus the mass of water for each peptide bond in the peptide.

MySQL

MySQL is an open-source relational database management system (RDBMS) created by a Swedish company, MySQL AB.

NCBI

National Center for Biotechnology Information (NCBI) is a US government resource that develops, distributes, supports, and coordinates access to a variety of databases and software for the scientific and medical communities.

NMR spectroscopy

Nuclear magnetic resonance (NMR) spectroscopy is a technique used to determine the structure of a compound.

PDB

Protein Data Bank (PDB) is a repository of curated 3D biological structures, usually proteins, populated by direct input from an international community of scientists. PDB utilizes different file formats (mmCif/XML) to display protein structure information.

Peptide

A small protein usually less than 100 amino acids in length.

Perl

Perl (short for "Practical Extraction and Report Language") is a highly capable, feature-rich programming language with over 29 years of development. Perl runs on over 100 platforms from portables to mainframes and is suitable for both rapid prototyping and large scale development projects.

Phi/Psi

Phi and Psi angles are the torsional angles around the backbone of an amino acid within a protein.

Pre- / Pro-

Prefixes in protein terminology that indicate various nascent forms of the peptide prior to activation. Pro- refers to an inactive peptide that requires additional post-transcriptional modifications prior to activation. Pre- indicates that an amino acid sequence requires additional cleavage prior to generating an active peptide.

Secondary structure

The secondary structures are the localized arrangements amino acids within a peptide resulting from interactions between the amino acid side-chains as well as between the side-chains and the environment.

Side-chains

Side-chains are the unique to each amino acid and give the amino acid unique chemical and physical properties.

Virtual Server

A virtual server is software that functions as if it were a physical server. It allows for easy mobility that a physical server could not provide.

XML

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable. The design goals of XML emphasize simplicity, generality and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages.

X-ray crystallography

X-ray crystallography is a technique used to determine the structure of a compound.

APPENDIX A

Chart matching the AMPed and PDB tables

AMPed Tables		PDB tables	
Peptide		struct_ref	entity
<i>Attribute</i>		biol_id	details
AMP_ID		db_code	formula_weight
Accession_No		db_name	id
Name		entity_id	pdxb_description
AA_sequence		id	pdxb_number_of_molecules
Mol_weight		pdxb_align_begin	src_method
Notes		pdxb_db_accession	type
Length_seq		pdxb_seq_one_letter_code	
		entity_poly	pdxb_database_related
		entity_id	content_type
		nstd_linkage	db_id
		nstd_monomer	db_name
		pdxb_seq_one_letter_code	details
		pdxb_seq_one_letter_code_can	
		pdxb_strand_id	
		type	
3D_Structure		entity	
<i>Attribute</i>		details	
AMP_ID		formula_weight	
PDB_ID		id	
Source_ID		pdxb_description	
		pdxb_number_of_molecules	
		src_method	
		type	
Amino_Acid_Address		entity_poly_seq	entity_poly
<i>Attribute</i>		entity_id	entity_id
AMP_ID		hetero	nstd_linkage

AA_Names	mon_id	nstd_monomer
Chain	num	pdxb_seq_one_letter_code
2nd_structure		pdxb_seq_one_letter_code_can
Sequence_AA_No		pdxb_strand_id
hydro	pdxb_validate_torsion	type
Phi	auth_asym_id	
Psi	auth_comp_id	
	auth_seq_id	
	id	
	PDB_ins_code	
	PDB_model_num	
	phi	
	psi	
Atom_coord_Source	exptl	
Attribute	crystals_number	
Source_ID	entry_id	
Source_Name	method	
Description		
Atomic_Coordinates	atom_site	entity_poly_seq
Attribute	auth_asym_id	entity_id
AA_Names	auth_atom_id	hetero
Atom_num	auth_comp_id	mon_id
Atom_Names	auth_seq_id	num
X	B_iso_or_equiv	
Y	B_iso_or_equiv_esd	
Z	Cartn_x	
Atom_Types	Cartn_x_esd	
Error	Cartn_y	
Sequence_AA_No	Cartn_y_esd	
Chain	Cartn_z	
	Cartn_z_esd	
	group_PDB	
	id	
	label_alt_id	

		label_asym_id		
		label_atom_id		
		label_comp_id		
		label_entity_id		
		label_seq_id		
		occupancy		
		occupancy_esd		
		pdbx_formal_charge		
		pdbx_PDB_ins_code		
		pdbx_PDB_model_num		
		type_symbol		
Gene		entity_src_nat		
Attribute		common_name		
Genome_ID		details		
Genome_Name		entity_id		
DNA_seq		genus		
chromosome_num		pdbx_atcc		
chrom_address_start		pdbx_cell		
chrom_address_end		pdbx_cell_line		
Species		pdbx_cellular_location		
RNA_Sequence		pdbx_fragment		
DNA_ID		pdbx_ncbi_taxonomy_id		
RNA_ID		pdbx_organ		
AMP_ID		pdbx_organelle		
		pdbx_organism_scientific		
		pdbx_plasmid_details		
		pdbx_plasmid_name		
		pdbx_secretion		
		pdbx_variant		
		species		
		strain		
		tissue		
		tissue_fraction		
Article		citation		citation_author
Attribute		book_publisher		citation_id

AMP_ID		country	name
Abstract		id	ordinal
Doi		journal_abbrev	
Volume		journal_id_ASTM	
Year		journal_id_CSD	
Authors		journal_id_ISSN	
Article_Title		journal_volume	
Journal_Article		page_first	
Start_Page		page_last	
Start_End		pdxb_database_id_DOI	
Pages	*	pdxb_database_id_PubMed	
Article_ID		title	
		year	

* = 1 + (_citation.page_last - _citation.page_first)

APPENDIX B

Database Structure and Schema of AMPed

Database Structure

3D_Structure				
Field	Type	Key	Default	Extra
AMP_ID	varchar(15)	NO		
PDB_ID	varchar(10)	NO	PRI	
Source_ID	varchar(5)	YES		
Resolution	Decimal(5,3)	NO		

Access_Level				
Field	Type	Key	Default	Extra
Access_Level	int(2)	NO	PRI	0
Description	varchar(20)	YES		

Amino_Acid_Address				
Field	Type	Key	Default	Extra
AMP_ID	varchar(15)	NO		
Phi	int(3)	YES		
Psi	int(3)	YES		
2nd_structure	text	YES		
hydro	text	YES		
Chain	varchar(5)	NO		
Sequence_AA_No	int(10)	NO		
AA_Symbol	char(1)	NO		

Article				
Field	Type	Key	Default	Extra
AMP_ID	varchar(15)	NO		
Abstract	text	YES		
Doi	text	YES		
Volume	varchar(20)	YES		
Year	year(4)	YES		
Authors	text	YES		
Article_Title	varchar(100)	YES		
Pages	int(5)	YES		
Journal_title	varchar(100)	YES		

Start_Page	int(10)	YES		
End_Page	int(10)	YES		
Article_ID	varchar(5)	NO	PRI	

Atom_coord_Source				
Field	Type	Key	Default	Extra
Source_Name	varchar(20)	YES		
Description	text	YES		
Source_ID	varchar(5)	NO	PRI	

Atomic_Coordinates				
Field	Type	Key	Default	Extra
X	decimal(6,3)	YES		
Y	decimal(6,3)	YES		
Z	decimal(6,3)	YES		
AA_Symbol	char(1)	NO		
Error	decimal(6,2)	YES		
Atom_num	int(10)	YES		
Atom_Names	varchar(10)	YES		
Atom_Types	varchar(10)	YES		
Sequence_AA_No	int(10)	NO		
Chain	varchar(5)	NO		

Country				
Field	Type	Key	Default	Extra
Country_ID	varchar(3)	NO	PRI	
Country	varchar(40)	YES		

Fight_Against				
Field	Type	Key	Default	Extra
AMP_ID	varchar(15)	NO		
Microbe_ID	varchar(20)	NO	MUL	
ATCC_ID	int(11)	YES		

Gene				
Field	Type	Key	Default	Extra
DNA_Seq	text	YES		
chromosome_num	varchar(255)	YES		
Species	text	YES		
RNA_Sequence	text	YES		

Genome_ID	int(11)	YES		0
chrom_address_start	decimal(30,20)	YES		
chrom_address_end	decimal(30,20)	YES		
Genome_Name	text	YES		
DNA_ID	varchar(20)	NO	PRI	
RNA_ID	varchar(20)	YES		
AMP_ID	varchar(15)	YES		

Inserted_By				
Field	Type	Key	Default	Extra
AMP_ID	varchar(15)	NO	PRI	
UserName	varchar(45)	YES		
Bioparser	int(11)	YES		0
Approval_Status	varchar(20)	YES		

Method				
Field	Type	Key	Default	Extra
Method_Name	varchar(20)	NO		
Description	text	YES		
Method_ID	varchar(10)	NO	PRI	

Microbe				
Field	Type	Key	Default	Extra
Species_Name	text	YES		
Microbe_Type	varchar(50)	YES		
Microbe_ID	varchar(20)	NO	PRI	

Peptide				
Field	Type	Key	Default	Extra
AMP_ID	varchar(15)	NO	PRI	
Accession_No	varchar(15)	NO	MUL	
Name	text	YES		
Mol_Weight	varchar(15)	YES		
AA_Sequence	text	YES		
Notes	text	YES		
Length_seq	int(10)	YES		

Results_of_Test				
Field	Type	Key	Default	Extra
Microbe_ID	varchar(20)	NO		

Test_ID	varchar(10)	NO
---------	-------------	----

Test				
Field	Type	Key	Default	Extra
MIC	decimal(10,0)	YES		
UOM	varchar(20)	YES		
Method_ID	varchar(20)	YES		
Microbe_ID	varchar(20)	YES		
Test_ID	varchar(10)	NO	PRI	

Used_Method				
Field	Type	Key	Default	Extra
Test_ID	varchar(10)	NO		
Method_ID	varchar(10)	NO		

User				
Field	Type	Key	Default	Extra
UserName	varchar(45)	NO	PRI	
Password	varchar(20)	NO		
Access_Level	int(2)	YES		
First_Name	varchar(20)	NO		
Last_Name	varchar(20)	YES		
Email	varchar(45)	YES		
Contact_No	varchar(20)	YES		
Job_Title	varchar(20)	YES		
Affiliation	varchar(50)	YES		
Country_ID	varchar(5)	YES		
Address	varchar(50)	YES		
Building	varchar(20)	YES		
City	varchar(20)	YES		
State	varchar(20)	YES		
Zip_Code	varchar(15)	YES		
Note	text	YES		

Table 1: AMPed tables with attributes and data type

The references about these tables are as follows:

- **Field** indicates the table column name.

- **Type** indicates the column data type: int, varchar, text, etc.
- **Key** fields do not permit NULL values as indicated by “NO” in the Key column.
- The **Default** field indicates whether the column is indexed (an indexed column stores copies of the values from that column in a data structure, allowing fast lookups for the rows with the corresponding column values):
 - If empty, the column is either not indexed or is indexed only as a secondary column in a multiple-column, nonunique index.
 - If **PRI**, the column is a PRIMARY KEY or is one of the columns in a multiple-column PRIMARY KEY.
 - If **UNI**, the column is the first column of a UNIQUE index. (A UNIQUE index permits multiple NULL values, but you can tell whether the column permits NULL by checking the Null field.)
 - If **MUL**, the column is the first column of a nonunique index in which multiple occurrences of a given value are permitted within the column.
- The **Extra** field contains any additional information that is available about a given column.

Database Schema

Over the past three years the AMPed structure has suffered many modifications designed to normalize the database and to improve its functionality. Figure 40 shows the database structure version 1.0 as designed by George Konstantinidis. Figure 41 shows the DBS version 2.0 as improved by Tripti Garg, while figure 42 displays the current version, 3.0 improved during this thesis.

Figure 40: AMPed structure version 1.0

Figure 41: AMPed structure version 2.0

Figure 42: AMPed structure version 3.0

APPENDIX C

Comparison between AMPed and other antimicrobial peptides repositories online

The table below, originally collected by Tripti Garg and updated for this project compares other antimicrobial peptides repositories online with URI's AMPed's site. The websites being compared are:

- The Antimicrobial Peptide Database (APD).
<http://aps.unmc.edu/AP/main.php>
- Yet another db of antimicrobial peptides (YADAMP).
<http://yadamp.unisa.it/>
- A database linking antimicrobial peptide (LAMP).
<http://biotechlab.fudan.edu.cn/database/lamp/>

Category	YADAMP	APD	LAMP	AMPed
Security				
Privacy	N/A	N/A	Store private information in cookie that is accessible	Cookie created but destroyed after the session
Password	No	No	No	Yes
CAPTCHA	No	No	No	Yes
Access Request	No	No	No	Yes
Secure access (HTTPS)	No	No	No	Yes

Category	YADAMP	APD	LAMP	AMPed
Design and Accessibility				
Physical limits of visual acuity	No	No	Yes	Yes
Limits of absolute memory	No	No	No	Yes
Gestalt principle	Yes	No	Yes	Yes
Category	YADAMP	APD	LAMP	AMPed
Website Features				
Persistent Navigation	Yes	No	Yes	Yes
Consistent Footer	Yes	No	Yes	Yes
SiteMap	No	No	No	Yes
Visible Branding	Yes	Yes, but not on every page	Yes	Yes
About the Team	No	No	Yes	Yes
Contact Us	Yes, e-mail only	Yes	Yes, e-mail only	Yes

Number of Visits	No	Yes	Yes	Yes
Track visits per session	No	No	No	Yes
Database Statistics	Yes	Yes	Yes	Yes
Own Domain	Yes	No	No	Yes
Favicon	No	Yes	No	Yes
Website's Admin email	No	No	No	Yes
Portability (desktops, tables, mobiles)	No	No	No	Yes
Category	YADAMP	APD	LAMP	AMPed
Website Data				
Peptides	Yes	Yes	Yes	Yes
Genomes	No	No	No	Yes
3D Structures	Yes	Yes	No	Yes
Amino Acids	No	No	No	Yes
Microbes	Yes	No	No	Yes
Test/Method	Yes	No	No	Yes
Atomic Coordinates	Yes	No	Yes	Yes
Source	Yes	Yes	Yes	Yes

MIC	Yes	Yes	Yes	Yes
Fight Against	Yes	Yes	Yes	Yes
Uniprot ID	Yes	No	Yes	Yes
Author	Yes	Yes	Yes	Yes
Length	Yes	Yes	Yes	Yes
Category	YADAMP	APD	LAMP	AMPed
Search				
Flexibility	High	High	Low	Medium
Visual Acuity	Low (search buttons are very small)	Low	Low	High
Number of Search Criterion	26	14	10	6
Summary Results	Yes	Yes	Yes	Yes
Detailed Results	Yes	Yes	Yes	Yes
Number of Results Returned	Yes	Yes	No	Yes
Search Criteria for Peptide Sequence	Yes	Yes	No	Yes, both Partial and Full Search

**Table 2: AMPed comparison with other Database repositories
(Courtesy of Tripti Garg and enhanced during this project)**

APPENDIX D

Perl code to read and parse a PDBML/XML file and output run

Main script for reading an XML file.

```
#!/usr/bin/perl

# -----
# Perl program to read and parse a PDBML/XML file:
# read-pdb-xml.pl file
# -----
# Written by: AbrAhAm HerrerA (2016)
# -----

# importing XML module
use XML::Simple;

# create object
$xml = new XML::Simple;

# call to additional perl script required to calculate the molecular weight
# by importing the weightMol() function
require "molWeight.pl";

# read single XML file; example:
# D:\wdir\URI\thesis\RCSB-PDB\AntiMPep-from-PDB-161017\1zmh.xml
$file = " AntiMPep-from-PDB-161017/1zmh.xml";
print "Reading file: $file.\n";
$data = $xml->XMLin($file);

# access XML data
print "Peptide table\n";
$AMP_ID = "AMP".$data->{datablockName};
print "AMP_ID: $AMP_ID\n";

$Accession_No = $data->{'PDBx:struct_refCategory'}->{'PDBx:struct_ref'}-
>{'PDBx:pdxb_db_accession'};
print "Accession_No: $Accession_No\n";

$Name = $data->{'PDBx:structCategory'}->{'PDBx:struct'}->{'PDBx:pdxb_descriptor'};
print "Name: $Name\n";
```

```

$AA_sequence = $data->{'PDBx:entity_polyCategory'}->{'PDBx:entity_poly'}-
>{'PDBx:pdxb_seq_one_letter_code_can'};
print "AA_sequence: $AA_sequence\n";

$Mol_weight = weightMol($AA_sequence);
print "Mol_weight: $Mol_weight\n";

$Notes = $data->{'PDBx:pdxb_database_relatedCategory'}-
>{'PDBx:pdxb_database_related'}[0]->{'PDBx:details'};
print "Notes: $Notes\n";

$Length_seq = length $AA_sequence;
print "Length_seq: $Length_seq\n";
print "\n";

print "3D_Structure table\n";
print "AMP_ID: $AMP_ID\n";

$PDB_ID = $data->{'PDBx:entryCategory'}->{'PDBx:entry'}->{'id'};
print "PDB_ID: $PDB_ID\n";
print "\n";

print "Amino_Acid_Address table\n";
print "AMP_ID: $AMP_ID\n";

$AA_Names = $data->{'PDBx:entity_poly_seqCategory'}->{'PDBx:entity_poly_seq'}[0]-
>{'mon_id'};
print "AA_Names: $AA_Names\n";

$Chain = $data->{'PDBx:entity_polyCategory'}->{'PDBx:entity_poly'}-
>{'PDBx:pdxb_strand_id'};
print "Chain: $Chain\n";

print "Length_seq: $Length_seq\n";

$Sequence_AA_No = $data->{'PDBx:entity_poly_seqCategory'}-
>{'PDBx:entity_poly_seq'}[0]->{'num'};
print "Sequence_AA_No: $Sequence_AA_No\n";

$Phi = $data->{'PDBx:pdxb_validate_torsionCategory'}->{'PDBx:pdxb_validate_torsion'}-
>{'1'}->{'PDBx:phi'};
print "Phi: $Phi\n";
$Psi = $data->{'PDBx:pdxb_validate_torsionCategory'}->{'PDBx:pdxb_validate_torsion'}-
>{'1'}->{'PDBx:psi'};

```



```

print "Psi: $Psi\n";
print "\n";

print "Atomic_Coordinates table\n";
$numAtomCoords = scalar keys(%{$data->{'PDBx:atom_siteCategory'}-
>{'PDBx:atom_site'}});
print "MULTIPLE ENTRIES IN DATABASE (Total # of entries: $numAtomCoords)\n";
print "Below is an example; entries 1 -- 10\n";
for (my $id = 1; $id < 11; $id++) {
    $Atom_num = $id;
    print "Atom_num: $Atom_num\n";
    $AA_Names = $data->{'PDBx:atom_siteCategory'}->{'PDBx:atom_site'}->{$id}-
>{'PDBx:auth_comp_id'};
    print "AA_Names: $AA_Names\n";
    $Atom_Names = $data->{'PDBx:atom_siteCategory'}->{'PDBx:atom_site'}->{$id}-
>{'PDBx:type_symbol'};
    print "Atom_Names: $Atom_Names\n";
    $X = $data->{'PDBx:atom_siteCategory'}->{'PDBx:atom_site'}->{$id}-
>{'PDBx:Cartn_x'};
    print "X: $X\n";
    $Y = $data->{'PDBx:atom_siteCategory'}->{'PDBx:atom_site'}->{$id}-
>{'PDBx:Cartn_y'};
    print "Y: $Y\n";
    $Z = $data->{'PDBx:atom_siteCategory'}->{'PDBx:atom_site'}->{$id}-
>{'PDBx:Cartn_z'};
    print "Z: $Z\n";
    $Atom_Types = $data->{'PDBx:atom_siteCategory'}->{'PDBx:atom_site'}->{$id}-
>{'PDBx:label_atom_id'};
    print "Atom_Types: $Atom_Types\n";
    $Error = $data->{'PDBx:atom_siteCategory'}->{'PDBx:atom_site'}->{$id}-
>{'PDBx:B_iso_or_equiv'};
    print "Error: $Error\n";
    print "-----\n";
}
print "\n";

print "Gene table\n";

print "AMP_ID: $AMP_ID\n";
print "\n";

print "Article table\n";
print "AMP_ID: $AMP_ID\n";

```

```

$Doi = $data->{'PDBx:citationCategory'}->{'PDBx:citation'}-
>{'PDBx:pdxb_database_id_DOI'};
print "Doi: $Doi\n";
$Volume = $data->{'PDBx:citationCategory'}->{'PDBx:citation'}-
>{'PDBx:journal_volume'};
print "Volume: $Volume\n";
$Year = $data->{'PDBx:citationCategory'}->{'PDBx:citation'}->{'PDBx:year'};
print "Year: $Year\n";

@allAuthors = keys(%{$data->{'PDBx:citation_authorCategory'}-
>{'PDBx:citation_author'}});
# initializing 'authOrd' hash table
%authOrd;
foreach my $thisAuthor (@allAuthors) {
    $order = $data->{'PDBx:citation_authorCategory'}->{'PDBx:citation_author'}-
>{$thisAuthor}>{'ordinal'};
    $authOrd{$thisAuthor} = $order;
}
@keys = sort { $authOrd{$a} <=> $authOrd{$b} } keys(%authOrd);
@vals = @authOrd{@keys};

# joining list of authors in a single line
$Authors = join(' ', @keys);
print "Authors: $Authors\n";

$Article_Title = $data->{'PDBx:citationCategory'}->{'PDBx:citation'}->{'PDBx:title'};
print "Article_Title: $Article_Title\n";

$Journal_Article = $data->{'PDBx:citationCategory'}->{'PDBx:citation'}-
>{'PDBx:journal_abbrev'};
print "Journal_Article: $Journal_Article\n";

$Start_Page = $data->{'PDBx:citationCategory'}->{'PDBx:citation'}->{'PDBx:page_first'};
print "Start_Page: $Start_Page\n";

$Start_End = $data->{'PDBx:citationCategory'}->{'PDBx:citation'}->{'PDBx:page_last'};
print "Start_End: $Start_End\n";
print "\n";

print "Inserted_By table\n";
print "AMP_ID: $AMP_ID\n";
print "Bioparser: 1\n";
print "Approval_Status: Approved\n";

```

Required Perl script needed for calculating the molecular weight of a sequence (

```
# -----  
# Perl program which reads calculates the weight of a sequence  
#  
# -----  
# Written by: AbrAhAm HerrerA (2016)  
# -----  
sub weightMol {  
  
    my ($string) = @_;  
  
    # initializing total sum  
    $sum = 0;  
  
    # removing carriage return if any  
    chomp($string);  
  
    # iterating from sequence to add their molecular weight  
    foreach $char (split //, $string) {  
        # collecting the weight in the variable $sum  
        $sum += $molWeight{$char};  
    }  
  
    # adding the hydrogen end to the sum  
    #print "Adding hydrogen end value ($hydroEnd) to this total\n\n";  
    $sum += $hydroEnd;  
  
    # return final addition  
    return $sum;  
} # ends subroutine  
  
# hash table with molecular weights  
%molWeight = (  
    A => 71.0782,  
    C => 103.143,  
    D => 115.0877,  
    E => 129.1143,  
    F => 147.1741,  
    G => 57.0516,  
    H => 137.1396,  
    I => 113.1579,  
    K => 128.1726,
```

```

L => 113.1579,
M => 131.196,
N => 114.1029,
P => 97.1155,
Q => 128.1295,
R => 156.186,
S => 87.0776,
T => 101.1042,
V => 99.1313,
W => 186.2102,
Y => 163.1735,
);
# hydrogen end value
$hydroEnd = 18.015;

```

Output from running the main Perl script.

```

D:\wdir\URI\thesis\RCSB-PDB>perl read-pdb-xml.pl
Reading file: AntiMPep-from-PDB-161017/1zmh.xml.

```

Peptide table

```

AMP_ID: AMP1ZMH
Accession_No: P59666
Name: Neutrophil defensin 2
AA_sequence: CYCRIPACIAGERRYATCIYQGRLWAFCC
Mol_weight: 3391.0329
Notes: Crystal structure of related defensin
Length_seq: 29

```

3D_Structure table

```

AMP_ID: AMP1ZMH
PDB_ID: 1ZMH

```

Amino_Acid_Address table

```

AMP_ID: AMP1ZMH
AA_Names: CYS
Chain: A,B,C,D
Length_seq: 29
Sequence_AA_No: 1
Phi:
Psi:

```

Atomic_Coordinates table

MULTIPLE ENTRIES IN DATABASE (Total # of entries: 1099)

Below is an example; entries 1 -- 10

Atom_num: 1
AA_Names: CYS
Atom_Names: N
X: 2.916
Y: 0.742
Z: 1.707
Atom_Types: N
Error: 22.08

Atom_num: 2
AA_Names: CYS
Atom_Names: C
X: 4.174
Y: 0.055
Z: 1.212
Atom_Types: CA
Error: 20.52

Atom_num: 3
AA_Names: CYS
Atom_Names: C
X: 4.796
Y: -0.780
Z: 2.312
Atom_Types: C
Error: 18.76

Atom_num: 4
AA_Names: CYS
Atom_Names: O
X: 4.618
Y: -0.466
Z: 3.490
Atom_Types: O
Error: 19.14

Atom_num: 5
AA_Names: CYS
Atom_Names: C
X: 5.207
Y: 1.035
Z: 0.718

Atom_Types: CB
Error: 21.88

Atom_num: 6
AA_Names: CYS
Atom_Names: S
X: 4.711
Y: 2.136
Z: -0.633
Atom_Types: SG
Error: 25.15

Atom_num: 7
AA_Names: TYR
Atom_Names: N
X: 5.555
Y: -1.819
Z: 1.931
Atom_Types: N
Error: 19.49

Atom_num: 8
AA_Names: TYR
Atom_Names: C
X: 6.124
Y: -2.795
Z: 2.903
Atom_Types: CA
Error: 18.71

Atom_num: 9
AA_Names: TYR
Atom_Names: C
X: 7.462
Y: -3.278
Z: 2.366
Atom_Types: C
Error: 19.99

Atom_num: 10
AA_Names: TYR
Atom_Names: O
X: 7.547
Y: -3.654

Z: 1.190

Atom_Types: O

Error: 21.78

Gene table

AMP_ID: AMP1ZMH

Article table

AMP_ID: AMP1ZMH

Doi: 10.1074/jbc.M503084200

Volume: 280

Year: 2005

Authors: Xie, C.; Prahl, A.; Ericksen, B.; Wu, Z.; Zeng, P.; Li, X.; Lu, W.Y.; Lubkowski, J.; Lu, W.

Article_Title: Reconstruction of the conserved beta-bulge in mammalian defensins using D-amino acids.

Journal_Article: J.Biol.Chem.

Start_Page: 32921

Start_End: 32929

Inserted_By table

AMP_ID: AMP1ZMH

Bioparser: 1

Approval_Status: Approved

D:\wdir\URI\thesis\RCSB-PDB>

APPENDIX E

ER Diagram

An entity relationship (ER) diagram is a graphical way of displaying information on how an object relates within the system. The main components of the AMPed ER diagram are:

- Entities; represented by rectangles. An entity is an object or concept about which you want to store information.

- Actions, represented by diamond shapes, show how two entities share information in the database.

- Attributes, represented by ovals, are characteristic of the entity. A key attribute, represented with an underline, is a unique characteristic of an entity.

A multivalued attribute can have more than one value.

- Connecting lines are solid lines that connect attributes to show the relationships of entities in the diagram.
- Cardinality specifies how many instances of an entity relate to one instance of another entity.

Figure 43: AMPed ER diagram (High Definition)

BIBLIOGRAPHY

- [1] Antispam: Protect Your Website Against Spammers. URL: <http://antispam.byteplant.com/products/cleanmail/spambot-trap.html>, Last accessed: 10/05/2016
- [2] ATCC (American Type Culture Collection), URL: <http://www.atcc.org>, Last accessed: 10/05/2016
- [3] Free Website Terms and Conditions. URL: <http://www.contractology.com/free-website-terms-and-conditions.html>, Last accessed: 10/10/2016
- [4] G. Konstantinidis, J. Peckham, PhD, L. M. Martin, PhD. (2013). *"AMPed - Antimicrobial Peptide Editable Database, Approaches to a Collaborative Design of Bioinformatics Databases."* New York Structural Biology Discussion Group 8th Winter Meeting. New York Academy of Sciences, Thursday January 24.
- [5] G. Konstantinidis. (2014). *"Antimicrobial Peptide Editable Database"* (Master's thesis). Retrieved from <http://digitalcommons.uri.edu/cgi/viewcontent.cgi?article=1696&context=theses>
- [6] A. Izadpanah, RL. Gallo, *"Antimicrobial peptides"*. J Am Acad Dermatol. 2005 Mar; 52(3 Pt 1):381-90; quiz 391-2 [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/15761415>
- [7] J. Spurlock, *"Bootstrap: Responsive Web Development"*, First Edition, 2013
- [8] J. Valade, *"PHP and MySQL For Dummies"*, Fourth Edition, 2009

- [9] K. Noyes, "*Five Reasons Linux Beats Windows for Servers*." PCWorld. URL: http://www.pcworld.com/article/204423/why_linux_beats_windows_for_servers.html, Last accessed: 10/01/2016
- [10] L. Ullman, "*PHP for the World Wide Web*", Second edition, 2004
- [11] L. Wall, T. Christiansen, and J. Orwant, "*Programming Perl*", Third Edition, 2000
- [12] N. Patel, "*The 4 Most Important Pages on Your Website (& How to Optimize Them)*." URL: <http://blog.hubspot.com/marketing/optimize-important-website-pages>, Last accessed: 10/10/2016
- [13] P. J. Lynch and S. Horton, "*Web Style Guide*", Third edition, 2009
- [14] P. DiMaggio, E. Hargittai, W. Russell Neuman, and J. P. Robinson. (2001). "*Social Implications of the Internet*." [Online]. Available: <https://www.princeton.edu/~artspol/workpap/WP17%20-%20DiMaggio,%20Hargittai,%20Neuman,%20Robinson.pdf>
- [15] PDB (Protein Data Bank), URL: <http://www.rcsb.org>, Last accessed: 10/10/2016
- [16] R. Elmasri and S. B. Navathe, "*Fundamentals of Database Systems*", Seventh Edition, 2015. Pearson Publishers
- [17] R. Vellest, "*What's the point of favicons?*" URL: <http://www.webdesignerdepot.com/2012/11/whats-the-point-of-favicons/>, Last accessed: 10/01/2016
- [18] S. Cozens, "*Ten Perl Myths*." URL: <http://www.perl.com/pub/2000/01/10PerlMyths.html>, Last accessed: 10/01/2016

- [19] T. Converse and J. Park with C. Morgan, *"PHP5 and MySQL® Bible"*, Third edition, 2004
- [20] T. Garg. (2014). *"Graphical User Interface for Antimicrobial Peptide Database"* (Master's thesis). Retrieved from <http://digitalcommons.uri.edu/cgi/viewcontent.cgi?article=1836&context=theses>
- [21] T. Garg, G. Konstantinidis, J. Peckham, A. Monteiro, R. LaCroix, L. M. Martin. (2015). *"Designing AMPed, The Practical Antimicrobial Peptide Editable Database."* American Peptide Society. URL: <http://www.5z.com/24APS/24APS.2015.105>
- [22] T. Herrington, JD, PhD, "The Background of Social Issues Affecting Internet Users" in *Linking Russia and America in Web Forums: The Global Classroom Project*. Tech/Web, Eds. Michael Day and Carol Lipson. Erlbaum Publishers, 2005. pp. 167-192.
- [23] UniProt. URL: <http://www.uniprot.org/>, Last accessed: 10/10/2016