MAP BASED TEACHER CREDENTIALING IN THE UNITED STATES – CLIENT SIDE

A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Karuna V. Hotlani
Summer 2012
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Karuna V. Hotlani:

Map Based Teacher Credentialing in the United States – Client Side

Carl Eckberg, Chair
Department of Computer Science

Joseph Lewis
Department of Computer Science

Mary Ann Lyman-Hager
Department of European Studies

July 3, 2012
Approval Date
DEDICATION

I dedicate this thesis work to my dear parents, for their unconditional love, support and faith in me. Without their endless support it would have been a difficult task to achieve.
ABSTRACT OF THE THESIS

Map Based Teacher Credentialing in the United States – Client Side
by
Karuna V. Hotlani
Master of Science in Computer Science
San Diego State University, 2012

As we know there are 50 different states in USA, and each state has different requirements for teacher credentialing. Languages being taught in different states depend on the requirements of that specific state. Each state has different requirements to get credentialing in world languages.

For any aspiring teacher, it requires quality of effort and time to collect the teacher credentialing information for a particular state. This thesis provides a well-developed GUI based web tool (website) for prospective teaching applicants to obtain detailed teacher credentialing information. Being a map-based site, it provides a list of requirements with detailed procedural steps tailored to the user input. It manages user selection of world languages as well as procedural steps to gain a credential. The site describes how a user can get a credential in a specific world language with detailed specifics about specific exams, organizations that administer those exams. Credentialing in non-language areas is also covered.
# TABLE OF CONTENTS

PAGE

| ABSTRACT | ....................................................................................................................................... v |
| LIST OF FIGURES | ................................................................................................................. ix |
| ACKNOWLEDGEMENTS | .......................................................................................................... x |

## CHAPTER

1 INTRODUCTION .........................................................................................................1

1.1 Contribution .......................................................................................................2

1.2 Overview of the Thesis ......................................................................................3

2 LITERATURE REVIEW ..............................................................................................4

2.1 Teacher Credentialing ........................................................................................4

2.1.1 Teacher Certification ................................................................................4

2.1.2 University of Kentucky .............................................................................5

2.1.3 Certification Map ......................................................................................5

2.2 Development Challenges ...................................................................................6

2.3 Design Approach ...............................................................................................8

3 TECHNOLOGIES .........................................................................................................9

3.1 Motivation to Decide Technologies ...................................................................9

3.1.1 LAMP .......................................................................................................9

3.1.2 Advantages of LAMP ..........................................................................9

3.2 PHP ..................................................................................................................10

3.2.1 PHP Features and Advantages ................................................................10

3.2.1.1 PHP Features: .................................................................................11

3.2.1.2 PHP Advantages: ...........................................................................12

3.2.2 PHP Comparison with Other Languages ................................................12

3.3 HTML ..............................................................................................................13

3.4 Advantages of HTML ......................................................................................13

3.5 JavaScript .........................................................................................................14

3.6 jQuery ..............................................................................................................15

3.7 Advantages of jQuery ......................................................................................15
3.8 CSS .................................................................16
3.9 Advantages of CSS ........................................16

4 SYSTEM DESIGN ..................................................18
  4.1 User (Registered) Requirement ......................18
  4.2 User (Guest) Requirement ............................18
  4.3 System Requirement ......................................18
  4.4 System Modules ............................................20
    4.4.1 Register and Login Module ......................20
    4.4.2 Interface ..................................................20
    4.4.3 Social Connectivity .................................20
  4.5 Application Architecture ..............................20
    4.5.1 OOP .........................................................20
    4.5.2 Advantages of OOP: .................................21
    4.5.3 Three-tier Architecture .........................21
    4.5.4 Comparison with MVC Architecture ..........23

5 SYSTEM IMPLEMENTATION AND APPEARANCE ....25
  5.1 Method ..........................................................25
  5.2 Design Flow ...................................................25
    5.2.1 Registered User ....................................25
    5.2.2 Guest User .............................................26
    5.2.3 Registered User Account .......................26
    5.2.3 Registered User Account .......................29
    5.2.4 Registration ...........................................29
  5.3 Application Logic ...........................................34
    5.3.1 Google API ............................................34
    5.3.2 Facebook API ........................................35
    5.3.3 Clickable Map Logic ...............................36
  5.4 User Interface ................................................37
    5.4.1 Home Page ............................................38
    5.4.2 Registration Form .................................38
    5.4.3 Texas — Routes ....................................40
    5.4.4 Texas — Educator Preparation Program ....41
    5.4.5 Texas — Certification Area ....................42
5.4.6 Texas — Region .....................................................................................43
5.4.7 California — Credential........................................................................43
5.4.8 California — Subject ..............................................................................44
5.4.9 California — Examination ......................................................................45
5.4.10 California — Requirements ..................................................................46
5.5 System Security ...............................................................................................47
  5.5.1 Cryptography ..........................................................................................48
  5.5.2 OAuth ......................................................................................................49
  5.5.3 Beyond Client-Server ..............................................................................49
  5.5.4 OpenID ....................................................................................................50

6 CONCLUSION AND FUTURE ENHANCEMENT ...........................................51
  6.1 Conclusion ...................................................................................................51
  6.2 Future Enhancement ...................................................................................51
REFERENCES ...........................................................................................................52
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Teacher certification website.</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>University of Kentucky website.</td>
<td>6</td>
</tr>
<tr>
<td>2.3</td>
<td>Certification map website.</td>
<td>7</td>
</tr>
<tr>
<td>3.1</td>
<td>HTML DOM Tree.</td>
<td>14</td>
</tr>
<tr>
<td>4.1</td>
<td>Roles and responsibility.</td>
<td>19</td>
</tr>
<tr>
<td>4.2</td>
<td>Visual overview of three-tiered application.</td>
<td>22</td>
</tr>
<tr>
<td>5.1</td>
<td>Registered user flowchart.</td>
<td>27</td>
</tr>
<tr>
<td>5.2</td>
<td>Guest user flowchart.</td>
<td>28</td>
</tr>
<tr>
<td>5.3</td>
<td>Registered user account flowchart.</td>
<td>30</td>
</tr>
<tr>
<td>5.4</td>
<td>Registration flowchart.</td>
<td>31</td>
</tr>
<tr>
<td>5.5</td>
<td>Login with Google API.</td>
<td>34</td>
</tr>
<tr>
<td>5.6</td>
<td>Login with Facebook API.</td>
<td>36</td>
</tr>
<tr>
<td>5.7</td>
<td>Clickable image logic.</td>
<td>38</td>
</tr>
<tr>
<td>5.8</td>
<td>Home page</td>
<td>39</td>
</tr>
<tr>
<td>5.9</td>
<td>Registration form</td>
<td>39</td>
</tr>
<tr>
<td>5.10</td>
<td>Texas – Requirements</td>
<td>40</td>
</tr>
<tr>
<td>5.11</td>
<td>Texas – Traditional route</td>
<td>41</td>
</tr>
<tr>
<td>5.12</td>
<td>Texas – Educator preparation program.</td>
<td>41</td>
</tr>
<tr>
<td>5.13</td>
<td>Texas – Certification area</td>
<td>42</td>
</tr>
<tr>
<td>5.14</td>
<td>Texas – Region</td>
<td>43</td>
</tr>
<tr>
<td>5.15</td>
<td>Texas – Service centers</td>
<td>44</td>
</tr>
<tr>
<td>5.16</td>
<td>California – Prospective</td>
<td>44</td>
</tr>
<tr>
<td>5.17</td>
<td>California – Subjects</td>
<td>45</td>
</tr>
<tr>
<td>5.18</td>
<td>California – Examination</td>
<td>46</td>
</tr>
<tr>
<td>5.19</td>
<td>California – Requirements</td>
<td>47</td>
</tr>
<tr>
<td>5.20</td>
<td>California – Credential Routes</td>
<td>48</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

I would like to thank my project advisor, Dr. Mary Ann Lyman-Hager for giving me the opportunity to work on this thesis project. I am grateful for her continuous guidance and support throughout the project. I would also like to thank Mr Norman Leonard, Outreach director, and Ashley Wynne for giving their support in researching.

A special thanks to my Committee Chair, Dr. Carl Eckberg, for accepting me as his thesis student and for giving his endless support.

I am grateful to Dr. Joseph Lewis for being on my committee and for his help. Finally, I would like to express my sincere appreciation to all the people who have supported and encouraged me in completing this thesis project.

I am also grateful to Rahul Chaturvedi, thesis partner, for giving his endless support and help.
CHAPTER 1

INTRODUCTION

In the United States, certification is handled by the state Board of Education, and requirements vary from state to state. A teaching credential is a type of certification. Teachers interested in working with specific subjects such as special education, math, art, music, (at certain grade levels i.e. high school), may need to earn subject-specific teaching credentialing either prior to or within a time limited period of initiation [1]. But all public school teachers need to be credentialed.

The Language Acquisition Resource Center (LARC) is one of fifteen national Language Resource Centers funded through the U.S. Department of Education. It is very interested in facilitating the credentialing of teachers, particularly those of world languages that are critically needed in our society to encourage trade and international business, security, and well being of American citizens. This project responds to the requirements articulated by Dr Lyman-Hager, LARC Director and Mr Norman Leonard, Outreach director, who is also an experienced K-12 world language teacher.

A United States teacher credential offered in multiple or single subject, is obtained upon completion of a bachelor’s degree and prescribed professional education requirements. Teaching credentials are required in the United States to qualify for teaching in public schools as well as numerous other institutions. Requirements vary from state to state [2].

Teacher credentials are issued at state level, and they may or may not be transferable through a reciprocity agreement. Teacher credentials obtained in a specific state may be accepted by another without reciprocity. Some will accept teaching credentials obtained in other states subject to additional requirements [1].

Different states can be idiosyncratic in their requirements for new teachers, so one must make sure any program under consideration will qualify the applicant to work where he or she eventually wants to teach. If the applicant cannot find an online program that qualifies under state criteria, or is difficult to arrange, an on-campus certification program in a specific geographic area would be a better option [3].
At present, obtaining detailed requirements and procedural steps to gain teacher credentials is very challenging for prospective students. There exists no single interface for different state requirements or different requirements for world languages. A few websites are available however they not easily useable for a non-technical user. In addition, data provided is in a form which creates a cumbersome user experience.

As requirements differ from state to state for each credential category, steps required to gain a particular credential are co-dependent upon other parameters of the search.

Different categories of requirements are:

- Out of State
- Out of United State
- Single Subject
- Multiple Subject
- Peace Corps
- Private School
- In State
- Traditional Route
- Alternative Route

This thesis provides a well-developed GUI based web tool (website) for prospective teaching applicants to obtain detailed teacher credentialing information.

Being a map-based site, it provides a list of requirements with detailed procedural steps tailored to the user input. It manages user selection of world languages as well as procedural steps to gain a credential. The site allows the user to get a credential in a specific world language with details about specific exams, organizations which administer those exams, etc. It shows requirements for the traditional credentialing routes and alternative credentialing (non traditional) routes.

1.1 Contribution

I and another thesis candidate created a complete user interface to provide information relevant to teacher credentialing in two states in the United States (Texas and California). The user interface will provide information related to requirements for different categories and information related to different type of examinations for teacher credentialing. The website allows a user to create an account and thereby provide access to restricted
features. In addition to this, we are providing the user an interactive map to access different states and related information. This website also provide features “Login with Facebook” and “Login with Google”, using which the user can connect with social networking and use Facebook/Google credential to login to this website.

1.2 OVERVIEW OF THE THESIS

The report is organized as follows:

- Chapter 2: This chapter comprises of the literature review. It would provide references to the history of the thesis issue.
- Chapter 3: The technologies used to develop the system and the reasons for using them are described in this chapter.
- Chapter 4: System Design. System architecture and design are discussed in this chapter.
- Chapter 5: Implementation details of this system. This chapter also provides a brief description of the User Interface. Also this chapter explains system security techniques used in implementing this system.
- Chapter 6: Conclusion, which includes summary of findings and suggestions for improvement for this system.
CHAPTER 2

LITERATURE REVIEW

The relationship between good teaching and student achievement is one that elevates the importance of teacher quality in the eyes of parents, educators, and policy makers. According to 55% of American parents surveyed in 1998 by Louis Harris and Associates [4], the quality of teachers is “the greatest influence on student learning.” We know from research that good teaching does not happen by accident. While some teachers may have a special gift to help students learn, good teaching encompasses critical elements, such as knowledge of the learning process, child development, teacher experiences, academic ability, and content knowledge. Knowing about these critical elements and having the ability to contextualize them to the learner makes learning a much more meaningful and relevant pursuit, as well as that teacher ideal [4].

2.1 TEACHER CREDENTIALING

During design and implement phase of this project, we researched different websites that provide similar information. We researched a few websites which provide comprehensive teacher credentialing information for different states, to get a first hand user perspective about those websites. We based our assessment from the point of view of an aspiring teacher seeking to collect teacher credential information. Our primary focus was on whether the user would be able to obtain all relevant information and if the site was user friendly.

We have summarized a list of websites visited and our perspective on user experience challenges.

2.1.1 Teacher Certification

This website provides Teacher Credentialing information for all states of the United States. As shown in Figure 2.1 [5], the user interface of this website is not very attractive; the user interface of any web site should be interactive and user friendly. The site provides very
limited information about all state teacher credentialing information. It excludes information such as subjects and programs needed by a given state, as well as information about teacher credentialing. Even more so, the site does give details on exam requirements. Thus, a number of user questions remain unanswered such as “Which exam should be taken by aspiring teacher to get credentialing in a particular subject?”, “Where those exams should be taken?” and “Who administers those exams?” It is therefore concluded the site does not provide comprehensive information.

### 2.1.2 University of Kentucky

As seen in Figure 2.2 [6], the user interface of this website is minimalist and does not provide much information. As stated on the website, it provides teacher credentialing information for all 50 states of United States. In this website as the user clicks on any state name, it redirects the user to respective state’s board of education website, and then users have to search through the state’s board of education’s website to collect information.

### 2.1.3 Certification Map

As shown in Figure 2.3 [7], the Certification Map website has a good, attractive user interface on the home page, having has a national map with ability to click through to
individual states. So users can click on any state to get teacher-credentialing information for that state. For each state it provides brief information about prerequisites for teacher credentialing, preparation for teacher credentialing, and brief information of reciprocity. It does not provide detailed information about different routes, categories, requirements for each category, detailed program lists, detailed subject lists, examination details for each program/subject, information for world language credentialing, etc.

To summarize our search, we did not find any website with functionality to provide all features required by LARC:

- User friendly and attractive User Interface
- Detailed information about each state.
- Lists of program and subjects required for credentialing
- Details of different examination for each subject.
- Detailed information for world language credentialing along with core subjects.

### 2.2 Development Challenges

In this thesis implementation, we have provided data for the states of California and Texas. To get detailed information for both these states we researched through the Board Education website of the respective states, which are:

- California: http://www.ctc.ca.gov/credentials/requirements.html
- Texas: http://www.tea.state.tx.us/
Ashley Wynne, who works under Director of Language Acquisition Resource Center (LARC) Dr. Lyman-Hager, also provided us with some researched data for the states of California and Texas. We collected all these data together and then designed the flow of data in the website. Using this data we analyzed the existing system to decide on an optimum method for data display, such that the user is able to retrieve information quickly.

User state of residence is not relevant and for which state he/she is interested in for teacher credentialing. He/she should be able to get teacher-credentialing information for any state. We also considered credentialing for world languages, which were not covered properly in previous websites. We created different credential categories after analyzing whole dataset. During our research we found out that each state’s data structure is completely different. So it was our main consideration to understand both states’ data and to design both interfaces to be flexible enough that they support respective data structures strongly and represent data to users in a very detailed and easy way.
2.3 Design Approach

The way we approached this problem is to first develop a well-designed and organized user interface for users to interact with the system. After researching a few available websites, what we found is that providing an interactive and clickable map to our end user would be a convenient option, as the user will be able to choose and be redirected to the state he/she wants to go quickly by just one click on the map of United States. This will help in solving the problem of searching for the option on the website to move to different state requirements. In addition to this, we added one more feature to our interface - a “bread crumb” approach for the user to easily keep track of pages or path he/she followed to be on the current page. This will also help the user in directly clicking and redirecting to any previous page he/she touched while coming to the current page.

These days website users are facing one more kind of problem, where they have to create and manage or remember account and their credentials on every other website they visit or where they have to register with new id. To take care of this kind of issue we are providing our user with features such as Google and Facebook logins, in which they can technically use their Google and Facebook credentials to create an account on our website.

We have created a simple and user-friendly interface, where data is displayed in a well organized manner just like the way we are storing it. All the options and data related to that are easily viewable to our user. This was a simple effort as part of this thesis to make the life of the end user simpler by providing him/her with an efficient and user-friendly interface.
CHAPTER 3

TECHNOLOGIES

This chapter focuses on technologies, which have been used to implement this thesis.

3.1 MOTIVATION TO DECIDE TECHNOLOGIES

The most important motivation to decide technologies was LAMP.

3.1.1 LAMP

LAMP is an acronym for a solution stack of free, open source software, referring to the first letters of Linux, Apache HTTP Server, MySQL and PHP. These are the principal components with which to build a viable general-purpose web server.

The exact combination of software included in a LAMP package may vary, especially with respect to the web scripting software, as PHP may be replaced or supplemented by Perl or Python.

Web application development using the LAMP stack is the preferred option for developers for creating a stable, reliable, and highly efficient application. It is interesting to note that they were developed individually, and at no point during the development of the software did the developers think about creating them for combined use. But it was found that taken together, they offer an unbeatable stack of solution-driven technologies that support application servers [8].

3.1.2 Advantages of LAMP

- Easy to code.
  It ensures that coding is relatively bug free and does not have to go through an exhaustive and time-consuming process of fixing the bug.
- Easy deployment
  PHP is a standard Apache module. This makes it easier to deploy LAMP web application.
- Local Development
  Developers can build an application locally and then deploy it into the web.
- **Cost**
  
  Each layer in the LAMP stack is open source. Thus initial startup costs are cut down significantly.

- **Scalability**
  
  LAMP architecture is horizontally scalable. Each layer in the stack can grow on its own and is loosely coupled with the other. The shared nothing architecture of PHP where each requirement is completely distinct and separate from any other requirement leads to infinite horizontal scalability in the language itself. PHP encourages you to push scalability issues to the layers that require it. It is also easy to maintain and thus to expand [9].

### 3.2 PHP

PHP is an HTML embedded scripting language. PHP stands for PHP: Hypertext Preprocessor. It originally stood for Personal Home Page.

PHP is general-purpose server side scripting language originally designed for web development to produce dynamic web pages. It is one of the first development server side scripting languages to be embedded into an HTML source document rather than calling an external file to process data. Ultimately a web server with a PHP processor module, which generates the resulting web page, interprets the code. It also has evolved to include a command-line interface capability and can be used in a standalone graphical application. PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform free of charge.

There are different versions of PHP available. This thesis has been developed using PHP version 5.3.1 [10].

Differences between earlier version of PHP and PHP version 5:

- New OOP model based on the Zend Engine 2.0
- New extension for improved MySQL support.
- Built-in native support for SQLite.
- New error reporting constant E STRICT, for runtime code suggestions.

A host of new functions to simplify code authoring [11].

### 3.2.1 PHP Features and Advantages

PHP features and advantages are explained below.
3.2.1.1 PHP Features:

- **Open Source**
  
  PHP is an open source language and is freely available for use. The community of open source PHP developers provides technology support and is constantly improving and updating the core PHP functionalities. The PHP extension and application repository system provides and maintain a library of PHP code packages that are available for use. The packages can include function such as authentication, caching, destructors, encryption, error handling etc.

- **Compatibility**
  
  PHP provides high compatibility with leading operating systems and web servers, thereby enabling it to be easily deployed across several different platforms.

- **File Handling**
  
  PHP be used to read text and generate files in various formats such as PDF and XML. Using the file manipulation function, files and documents can be uploaded and stored on the server. The uploaded documents can be accessed and manipulated through PHP code. PHP can be used to access flat files and to perform basic file and directory maintenance tasks, thereby enabling files/documents to be edited remotely.

- **Improved Performance**
  
  The PHP compiler includes features to optimize and improve the quality of compiled code by reducing the size execution time of the code, thereby leading to improved performance. The compiled PHP code can be cached using various PHP accelerators such as xcache, eAccelerator, etc. so as to reduce the time spend in parsing and compiling the code every time.

- **Debuggers**
  
  Several debuggers are available with PHP enabling developers to identify and analyze the code for potential bugs and bottlenecks.

- **Sessions**
  
  PHP provides extensive session and cookie management features and functions enabling the creation and development of personalized web page.

- **Graphics**
  
  PHP can be used to generate images and graphics dynamically using the image functionalities available with PHP, the header information of images can be accessed and manipulated. The GD library of PHP includes a host of features and functionalities that can be used to create images in various formats such as gif, jpeg and png.
    
    - **Extensible**
      
      The source code of PHP can be modified to include custom created extensions and components thereby increasing its extensibility.
3.2.1.2 PHP Advantages:

- PHP is available under the open source license, thereby making it a cost effective option.
- PHP speeds the load time and improves the browsing experience, as processing happens on the server.
- Major open source packages are written in PHP, thereby enabling easy customization and development.
- PHP is compatible with all major operating systems, web servers, and browsers.
- It includes a host of inbuilt features that simplify programming of common tasks.
- It is used in corporate websites, intranet applications and real estate portals [12].

3.2.2 PHP Comparison with Other Languages

We decided to use PHP by comparing it with other languages. Comparison of PHP with other languages is shown below.

- PHP vs. ASP

ASP is not really a language in itself but an acronym for Active Server Pages; the actual languages used to program. ASP includes Visual Basic Script, Jscript, and C# among others. The biggest drawback of ASP is that it is a proprietary system that is natively used only on the Microsoft Internet Information Server (IIS) platform. This limits its availability to Win32 based servers.

ASP is said to be a slower and more cumbersome language than PHP, with less overall stability. One of the pros of ASP is that, since it primarily uses VBScript, it is relatively easy to pick up the language if one is already proficient in Visual Basic. ASP support is also enabled by default in IIs, making it easy to get up and running. However the component built in ASP are really limited so if one needs to use “advanced” features, additional components will need to be purchased.

- PHP vs. ColdFusion

PHP is commonly said to be faster and more efficient for complex programming tasks and for trying out new ideas, and it is considered by many to be more stable and less resource intensive as well. While ColdFusion once had better error handling, database abstraction, and date parsing, the database abstraction shortfalls were addressed in PHP 4. A particular quality that is listed as one of ColdFusion’s strength is its excellent search engine, but many strong arguments suggest that a search engine is not something that should be included in a web scripting language. Further PHP runs on nearly every modern platform in existence, while ColdFusion is only available on Windows Solaris, Linux, Mac OS and A/X. ColdFusion has a good IDE and is generally quicker for a new programmer to achieve results for very simple applications, whereas PHP initially requires more programming knowledge. ColdFusion is designed with non-programmers in mind, while PHP is focused on programmers.
• **PHP vs. Perl**

The biggest advantage of PHP over Perl is that PHP was designed for scripting for the web, while Perl was designed to do a lot more. Because of this, Perl can get very complicated. The flexibility/complexity of Perl can make it difficult for developers of varying skill levels to collaborate. PHP has a less confusing and stricter format without losing flexibility. PHP is also easier to integrate into existing HTML than Perl. In large part, PHP has all the ‘good’ functionality of Perl construct, syntax, etc. without making its as complicated as Perl can be. Yet PHP’s command line interpreter is powerful enough to perform high-level tasks much in the same way Perl has been traditionally employed. Perl is a very tried and true language, and it has stood its ground since the 1980’s. On the other hand, PHP has matured and evolved quickly and continues to make fantastic progress[11].

### 3.3 HTML

HyperText Markup Language (HTML) is the main markup language for displaying web pages and other information that can be displayed in a web browser.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. Figure 3.1 [13] shows different tags and DOM tree of HTML.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. It can embed scripts in languages such as JavaScript, which affects the behavior of HTML webpages.

Web browsers can also refer to Cascading Style Sheet (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicitly presentational HTML markup.

### 3.4 ADVANTAGES OF HTML

- Hyper Text Markup Language (HTML) is a specific markup language defined in a few different versions by a standards body. It is the language that truly “builds” the website page with tables, divisions, and outputting text in certain formats, such as bold and underlined fonts.

- HTML presence is dominating the web, and it is the most widely accepted language for web design. It is easily recognized and interpreted by the popular web browsers.
- HTML is a very clean, straightforward language. This results in quicker download times and consistent viewing results. This is also very appealing to search engines.

- HTML allows programs to be embedded within it using an assortment of scripting languages. This results in a very flexible language. It is interactive with CSS, JavaScript and some Applets [14].

### 3.5 JavaScript

JavaScript is a prototype-based scripting language that is dynamic, weakly typed and has first class functions. It is a multi paradigm language, supporting object oriented, imperative, and functional programming styles.

JavaScript was formalized in the ECMA Script language standard and is primarily used in the form of client-side JavaScript. It is implemented as part of a web browser in order to give enhanced user interfaces and dynamic websites. This enables programmatic access to computational objects within a host environment.

JavaScript’s use in an application outside of web pages – for example in PDF documents, site-specific browser, and desktop widgets – is also significant. Newer and faster JavaScript VMs and frameworks built upon them have also increased the popularity of JavaScript for server-side web applications.
JavaScript uses syntax influenced by that of C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the Selfand Scheme programming languages [15].

3.6 JQUERY

jQuery is a cross-browser JavaScript library designed to simplify the client-side scripting of HTML. It was released in January 2006 at BarCamp NYC by John Resig. Used by over 55% of the 10,000 most visited websites, jQuery is the most popular JavaScript library in use today.

jQuery is free, open source software, dual-licensed under the MIT License or the GNU General Public License, Version 2. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop AJAX applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects, and high-level, theme-able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and web applications [16].

3.7 ADVANTAGES OF JQUERY

- Ease of use
  This is pretty much the main advantage of using jQuery, it is a lot more easy to use compared to standard JavaScript and other JavaScript libraries. Apart from simple syntax, it also requires much fewer lines of code to achieve the same feature in comparison.

- Large library
  jQuery enables one to perform hordes of function in comparison to other JavaScript libraries.

- Strong open source community
  jQuery, while relatively new, has a following that religiously devote their time to develop and enhance the functionality of jQuery. Thus there are hundreds of prewritten plugin available for download to instantly speed up the development process. Another advantage behind this is the efficiency and security of the script.
• Ajax support

jQuery lets one develop Ajax templates with ease. Ajax enables a sleeker interface where actions can be performed on pages without requiring the entire page to be reloaded [17].

3.8 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document, including plain XML, SVG and XUL.

There are many ways to link style sheets to HTML, each carrying its own advantages and disadvantages. New HTML elements and attributes have been introduced to allow easy incorporation of style sheets into HTML documents [18].

• Linking to an external style sheet:

The <link> tag is placed in the document head. The optional type attribute is used to specify a media type-text/css for a Cascading Style Sheet—allowing browsers to ignore style sheet types that they do not support. Configuring the server to send text/css as the content-type for CSS files is also a good idea.

• Embedding a style sheet:

The style element is placed in the document head. The required type attribute is used to specify a media type, as is its function with the link element. Similarly, the title and media attributes may also be specified with style.

• Importing a style sheet:

A style sheet may be imported with CSS's @import statement. Note that other CSS rules may still be included in the style element, but that all @import statements must occur at the start of the style sheet. Any rules specified in the style sheet itself override conflicting rules in the imported style sheets.

• Inline style

Style may be inlined using the style attribute. The style attribute may be applied to any body element (including body itself) except for basefont, param, and script. The attribute takes as its value any number of css declarations, where a semicolon separates each declaration [19].

3.9 ADVANTAGES OF CSS

• Consistency

By making one change to the website's CSS style sheet, one can automatically make it to every page of the website. The bigger the website, the more time CSS saves the
programmer. And not only does CSS save time, it also ensures that web pages have consistent styling throughout the site.

- **Bandwidth Reduction**

  When CSS separates the website's content from its design language, one dramatically reduces file transfer size. CSS document will be stored externally and will be accessed only once when a visitor requests the website. In contrast, when one creates a website using tables, every page of the website will be accessed with each visit. Reduced bandwidth needs will result in a faster load time and could cut web hosting costs.

- **Browser Compatibility**

  The recent arrival of Google® Chrome is further evidence that today's Internet users have more browser options than ever before, which makes browser compatibility a major issue for websites. CSS style sheets increase a website's adaptability and ensure that more visitors will be able to view the website in the way intended.

- **Viewing Options**

  Another common web design concern is the increasing need to make websites available for different media. CSS can help tackle this challenge by allowing the same markup page to be presented in different viewing styles - for example; one may create a separate style sheet for print or for a mobile device [20].
CHAPTER 4

SYSTEM DESIGN

This chapter discusses the system architecture used in this thesis. This chapter also explains about different types of user roles, requirements and different modules of code. The system requirements were similarly provided by

4.1 USER (REGISTERED) REQUIREMENT

- User must be able to register.
- User must be able to login with credentials.
- User must be able to use Facebook credentials to login.
- User must be able to use Google credentials to login.
- User must be able to click and search through on any state in USA map.
- User must be able to get information about teacher credentialing of any state.
- User should be able to edit account.
- User must be able to use forget password functionality.

4.2 USER (GUEST) REQUIREMENT

- User must be able to provide his own brief information.
- User must be able to click and search through on any state in USA map.
- User must be able to get information about teacher credentialing of any state.

4.3 SYSTEM REQUIREMENT

- There should be a clickable USA map for user flexibility.
- User Interface to display information about teacher credentialing of any state should be easily accessible.
- Information about teacher credentialing should be displayed separately for each state.
- Interface should display world language data in easily accessible form.

Figure 4.1 describes the use case diagram for the system. This diagram provides the graphical representation of all the entities (actors) and their roles. This diagram also elaborates interaction and dependencies between these entities.
Figure 4.1. Roles and responsibility.
4.4 SYSTEM MODULES

Different modules gather and create whole system. Those different modules of this thesis are:

4.4.1 Register and Login Module

- User can register to create an account.
- Registered user can get login with login credentials.
- Guest user can just provide his brief information without registering.
- Password is encrypted before storing in database.
- Forgot password asks user for email address and then it will send an email to user with his account information.

4.4.2 Interface

- User can click on any state in USA map to get information.
- Each state has different requirements and different teacher categories.
- Each state shows information about other subjects and world language.
- Each state provides the list of examination depending on selected world language or subject.
- Each state also provides detail about organization who administered examinations for teacher credentialing and also provides link to their online website.

4.4.3 Social Connectivity

- User can use his/her Facebook login credentials to get login into this website instead of creating a new account.
- User can use his/her Google login credentials to get login into this website instead of creating a new account.
- User can get connect with Facebook, Twitter, YouTube, RSS which gives ability to user to follow LARC on social networking.

4.5 APPLICATION ARCHITECTURE

This section explains architectural methods used to develop this system.

4.5.1 OOP

Object Oriented Programming is a programming paradigm using objects – data structures consisting of data fields and methods together with their interactions – to a design
application and computer programs. Programming techniques may include features such as
data abstraction, encapsulation, messaging, modularity, polymorphism, and inheritance.

Object-oriented programming has roots that can be traced to the 1960s. As hardware
and software became increasingly complex, manageability often became a concern.
Researchers studied ways to maintain software quality and developed object-oriented
programming in part to address common problems by strongly emphasizing discrete,
reusable units of programming logic. The technology focuses on data rather than processes,
with programs composed of self-sufficient modules (classes) each instance of which (objects)
contains all the information needed to manipulate its own data structure (members). This is in
contrast to the existing modular programming that had been dominant for many years that
focused on the function of a module, rather than specifically the data, but equally provided
for code reuse, and self-sufficient reusable units of programming logic,

This website is also implemented using object oriented programming. In application
logic tier of this system, for any logic it first creates controller class and use object of that
class anywhere in code. Using this methodology it separates logic modules and reuses those
modules whenever needed. This programming technique increases speed of compilation of
any logic.

4.5.2 Advantages of OOP:

- OOP provides a clear modular structure for programs which makes it good for
defining abstract data types where implementation details are hidden and the unit has
a clearly defined interface.
- OOP makes it easy to maintain and modify existing code as new objects can be
created with small differences to existing ones.
- OOP provides a good framework for code libraries where supplied software
components can be easily adapted and modified by the programmer. This is
particularly useful for developing graphical user interface [22].

4.5.3 Three-tier Architecture

Three-tier is a client-server architecture in which the user interface, functional process
logic (business rules), computer data storage and data access are developed and maintained
as independent modules, most often on separate platforms. It was developed by John J.
Donovan in Open Environment Corporation (OEC), a tools company he founded in Cambridge, Massachusetts.

The three-tier model is a software architecture and a software design pattern. Figure 4.2 [23] explains a three-tier architecture.


Apart from the usual advantages of modular software with well-defined interfaces, the three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently in response to changes in requirements or technology. For example, a change of operating system in the presentation tier would only affect the user interface code.

Typically, the user interface runs on a desktop PC or workstation and uses a standard graphical user interface, functional process logic may consist of one or more separate modules running on a workstation or application server, and an RDBMS on a database server or mainframe contains the computer data storage logic. The middle tier may be multi-tiered itself [23].
• Presentation tier

The presentation layer is a user interface of any application. It includes user controls and user input validation. This layer is also known as a thin client.

This thesis is mainly concentrated on implementing the presentation layer of the “Teacher Credentialing” website. Presentation layer of this website includes a clickable USA map, navigation on different pages, breadcrumbs, submenu of Texas, submenu of California. Presentation layer also consist login feature, registration form, user account page, etc.

• Application tier

The logic tier is pulled out from the presentation tier and, as its own layer, it controls an application’s functionality by performing detailed processing.

Application tier of this thesis represents the logic creating a new user, check login credential, edit or delete user account, forget password and sending mail functionality. Google API and Facebook API are also part of this tier.

• Data tier

This tier consists of database tables. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data on its own tier also improves scalability and performance.

4.5.4 Comparison with MVC Architecture

At first glance, the three tiers may seem similar to the model-view-controller (MVC) concept; however, topologically they are different. A fundamental rule in a three-tier architecture is the client tier never communicates directly with the data tier; in a three-tier model all communication must pass through the middle tier. Conceptually the three-tier architecture is linear. However, the MVC architecture is triangular: the view sends updates to the controller, the controller updates the model, and the view gets updated directly from the model.

From a historical perspective the three-tier architecture concept emerged in the 1990s from observations of distributed systems (e.g., web applications) where the client, middleware and data tiers ran on physically separate platforms. Whereas MVC comes from the previous decade and is based on observations of applications that ran on a single graphical workstation; MVC was applied to distribute applications later in its history.

Today, MVC and similar model-view-presenter (MVP) are separation of concerns design patterns that apply exclusively to the presentation layer of a larger system. In simple scenarios MVC may represent the primary design of a system, reaching directly into
the database; however, in most scenarios the Controller and Model in MVC have a loose dependency on either a Service or Data layer/tier [23].
CHAPTER 5

SYSTEM IMPLEMENTATION AND APPEARANCE

This chapter explains Graphical User Interface. It provides functional specification for each page with a description of how works. This chapter also includes the design flow of the whole system, application logic of the system and shows how these work functionally.

This chapter also covers security measures considered during implementation of this system.

5.1 Method

Chapter 2 describes issues and problematic areas with current systems, this chapter explains how those issues are getting solved in this system, and how this system has been designed. Chapter 4 explains how the three – tier architecture has been used in this system, which keeps the presentation layer separate from the logic layer, so in that way any changes in the user interface of the system does not affects the logic or functionality and vice-versa.

To implement this design structure, PHP was selected as the server side scripting language, using which function logic and database logic was implemented. To keep Graphical User Interface clean, HTML code was used along with CSS to keep the design and style of webpages more attractive and manageable. To add user flexibility and advanced effects in graphical user interface, jQuery and JavaScript were also used, along with HTML and CSS. MySQL was chosen to be data layer since it has well documented API towards PHP.

5.2 Design Flow

Different flow charts explain design flow and structure of different functionalities.

5.2.1 Registered User

Registered user is a feature provided by this system, when user wants to become a member of this website and wants to use other special features of this system, which are provided to registered users only. Special features are blog, update profile, connect with
social networking website, etc., along with searching through the website and getting information about teacher credentialing in those states.

The flowchart shown in Figure 5.1 explains the design flow and the process of this system for registered users. As a user comes to this website, he/she tries to login. If a user is registered with this website, then the user can use his/her login credentials to get login. If a user is not registered with this website he/she redirects to registration page and is asked to register first. Once a user has registered and has completed the login, he/she can click on any state from USA map on home page and go to any state. (Currently it only supports California and Texas States, but other states are under construction). After selecting any state, a user can search through that state and collect information about teacher credentialing, depending on his/her conditions. Users can also go to blog section and use blog facility. Users can also go in My Account and update his/her account detail.

5.2.2 Guest User

Guest user is a system feature, when user is a first time visitor of this website and is unsure of registering as a user. In this scenario, the user can search the website to get information about teacher credentialing in those states. Unregistered users do not have access to certain site functionality such as blog, update profile, get connect with social networking website, etc. Guest users can get register with this system at any stage and then will be able to use this special facilities provided by this system.

Flowchart shown in Figure 5.2 explains design flow and process of this system for a guest user. User experience can be described from the stage he visits the site. When the user opts for guest login, he is asked to provide registration information. This is brief and includes his first name, last name, email address, etc. After providing information, the user has access. He can view a state in the USA map on home page. (Currently it only supports California and Texas, but other states are under construction). After selecting a state, user can search and collect information about teacher credentialing depending on his/her conditions.

5.2.3 Registered User Account

A registered user has the ability to view his account, update it, or delete it whenever he/she wants to. The flowchart shown in Figure 5.3 explains the design flow of My Account page for registered users. As a registered user comes to this website, he/she logins with
Figure 5.1. Registered user flowchart.
his/her credentials. After getting logged in, users go to My Account page from navigation bar. On My Account page a user can view his account. User can edit any information of his profile, which gets update in database for respective user. User can also delete his account and deactivate his membership with this website. After deleting his account, he can only login is guest in future.
5.2.3 Registered User Account

A registered user has the ability to view his account, update it, or delete it whenever he/she wants to. The flowchart shown in Figure 5.3 explains the design flow of My Account page for registered users. As a registered user comes to this website, he/she logs in with his/her credentials. After getting logged in, users go to My Account page from navigation bar. On My Account page a user can view his account. User can edit any information of his profile, which gets update in database for respective user. User can also delete his account and deactivate his membership with this website. After deleting his account, he can only login is guest in future.

5.2.4 Registration

Users who want to register with this website have the option to use their Facebook/Google credentials to register. In that way users do not have to create new accounts and remember new user ids and passwords. Using this method a user can use his most used Facebook and/or Google credentials to login on this website.

Flowchart shown in Figure 5.4 explains the design flow for registration and social connectivity implemented in this website. As a user comes to this website and he/she wants to register with website, he/she gets three options: Normal registration, Login with Facebook and Login with Google. If user opts for normal registration he/she directly redirects to registration page. If user opts for Facebook login or Google login, he/she has to provide respective login credentials. In Login with Facebook, as user provides Facebook login credentials, Facebook server authenticates user and website and then returns with Facebook user id for respective user. In Login with Google, as user provides Google login credentials, Google server authenticates user and website and then returns with Google user id for respective user. Then User redirects to registration page of website. After filling out other information as user clicks on Register, website makes database entry with user’s information for login credential. If user has opted for Facebook or Google login, then Facebook user id or Google user id also gets enter into database table. The user is then redirected to the home page, where full site navigation and search may be repeated.
Figure 5.3. Registered user account flowchart.
Figure 5.4 Registration flowchart.
PROVIDE OTHER INFORMATION ON

MAKE DATABASE ENTRY FOR NEW USER WITH FACEBOOK/GOOGLE USERID

USER GETS LOGIN AND REDIRECT TO HOME PAGE AND WORK AROUND WEBSITE

END
5.3 APPLICATION LOGIC

This section explains the different logics used to develop this system.

5.3.1 Google API

This website has a special feature to ‘Login with Google’ during new user registration. A user who chooses not to register for any reason, can instead choose to use his/her Google credentials to create an account in this website. He/she can use Login with Google. Google API has been used to implement this feature in the website. Figure 5.5 [24] explains the flow of Google API to use Login with Google feature.

![Figure 5.5. Login with Google API. Source: Google Developers. (2012, June 26). Federated Login for Google Account Users [Online]. Available: https://developers.google.com/accounts/docs/OpenID#settingup.](image)

OpenID login authentication for web applications involves a sequence of interactions between your web application, Google's login authentication service, and the end user. The diagram and sequence in Figure 5.5 describe the process as recommended by Google.

- The user selects the "Sign in with Google" option.
- The web application sends a "discovery" request to Google to get information on the Google login authentication endpoint.
- Google returns an XRDS document, which contains the endpoint address.
- The web application sends a login authentication request to the Google endpoint address.
• This action redirects the user to a Google Federated Login page, either in the same browser window or in a popup window, and the user is asked to sign in.

• Once logged in, Google displays a confirmation page and notifies the user that a third-party application is requesting authentication. The page asks the user to confirm or reject linking their Google account login with the web application login. If the web application is using OpenID+OAuth, the user is then asked to approve access to a specified set of Google services. Both the login and user information sharing must be approved by the user for authentication to continue. The user does not have the option of approving one but not the other.

• If the user approves the authentication, Google returns the user to the URL specified in the openid.return_to parameter of the original request. A Google-supplied identifier, which has no relationship to the user's actual Google account name or password, is appended as the query parameter openid.claimed_id. If the request also included attribute exchange, additional user information may be appended. For OpenID+OAuth, an authorized OAuth request token is also returned.

• The web application uses the Google-supplied identifier to recognize the user and allow access to application features and data. For OpenID+OAuth, the web application uses the request token to continue the OAuth sequence and gain access to the user's Google services [24].

5.3.2 Facebook API

This website has a special feature to Login with Facebook during new user registration. If a user who does not want to create a new registration and username/password is willing to use his/her Facebook credentials to create account in this website, he/she can use Login with Facebook. Facebook An API has been used to implement this feature in the website. Figure 5.6 [25] explains the flow of Facebook API to use the Login with Facebook feature.

Facebook helps you simplify and enhance user registration and sign-in by using Facebook as your login system. Users no longer need to fill in yet another registration form or remember another username and password to use the site. As long as the user is signed into Facebook, they are automatically signed into this site as well. Using Facebook for login provides all the information needed to create a social, personalized experience from the moment the user visit your site in their browser. Facebook Platform uses OAuth 2.0 for authentication and authorization [25].
5.3.3 Clickable Map Logic

In this thesis, the USA map on the home page has a clickable state feature. Same in Texas state information region map has clickable region feature. To make any image clickable we need to find the x and y co-ordinates of that image. Those coordinates get use to find out other coordinates which finally define the whole image area which need to be click, all those coordinates get use in <area> tag in HTML code.
Below is example that shows how to calculate these coordinates.

In Figure 5.7 [26], the coordinates of woman are 15,48:50x133. These coordinates are the x:y coordinates of the upper left hand corner of the box and the x, y distances to the lower right hand corner. This coordinate system has its 0:0 point at the upper left corner of the image. So, the point 15:48 is 15 pixels from the left edge of the image, and 48 pixels from the top edge. The other two numbers indicate the location of the lower right corner in terms of pixels from the upper left corner. So, the lower right corner is 50 pixels to the right of the upper left corner and 133 pixels down from the upper left corner. The actual location of the lower right corner is 65,181. So clickable area is defined by 115, 48, 166, 221. [26]

5.4 USER INTERFACE

User interface is a presentation layer of three-tier architecture. It plays important role in any web or system application. It is an interface using that user interacts with any system. Interface should be user friendly and user flexible.

In this thesis user interface has been designed and implemented keeping non-technical user in consideration. It is flexible for any user that user can easily navigate from one page to other without getting confuse. For path track it also shows breadcrumbs of webpages, using which user can easily get where he is exactly in this website.

Next few figures shows screenshots of this website and explains each screenshot in detail.
5.4.1 Home Page

Figure 5.8 shows home page of the website. This page has a clickable USA map, where user can click on any state and get information about Teacher Credentialing.

This screen also provides login functionality. It has normal login and guest login. In login window user can input login credentials and get login to this website. This screen also provides social connectivity. Using “Login with Google” feature the user can use Google credentials to login and connect with this website. Using “Login with Facebook” feature the user can use Facebook credentials to login and connect with this website. A user can also get connect with social networking sites, like Facebook, Twitter, YouTube, RSS and follow LARC social networking.

5.4.2 Registration Form

Figure 5.9 shows Registration form of this website, which a user need to fill out and submit to get register with this website. As a user fill out this information and clicks on submit, new record gets insert into database for the new user with login credentials. On this screen a user provides password and that password gets encrypted before inserting into database.
If a user opts for “Login with Facebook” or “Login with Google” and as the user provides login credential for these web applications in return Teacher Credentialing website gets permission from Facebook / Google.

As this website gets permission from these web applications, it redirects the user to this registration form (shown in Figure 5.9) with prefilled basic information like First Name,
Last Name, Email, which has been returned from Facebook / Google. Then user fills out other information and create his/her account and gets login.

### 5.4.3 Texas — Routes

On Home Page, when user clicks on Texas State in USA map, the user redirects to Texas home page. Figure 5.10 shows main screen of Texas State.

![Texas State Home Page](image)

**Figure 5.10. Texas – Requirements.**

This figure shows sub menu of Texas, which includes:

- Requirements
- Educator Preparation Program
- Certification Area
- Region

Figure 5.10 shows two routes of teacher credentialing in Texas: the “Traditional” Route and the “Alternative” Route. As user clicks on any of these routes it shows steps to become teacher in Texas using that selected route. (See Figure 5.11.)
5.4.4 Texas — Educator Preparation Program

As a user selects Educator Preparation Program from submenu of Texas, a user gets list of Approved Program Entity available in Texas. Figure 5.12 shows the sample of Approved Program Entity list and how it is display on Teacher Credentialing website.

Figure 5.11. Texas – Traditional route.

Figure 5.12. Texas – Educator preparation program.
This page also provides feature of getting program list for world languages available in Texas and program list for core subjects available in Texas. User Interface of this page is designed in such way that user can easily navigate from Approved Program Entity List for World Languages to Approved Program Entity for Core Subjects. This thesis consider that Certification in a World Language may be added to certification in many states in a Core Subject area, and this website provides this feature.

5.4.5 Texas — Certification Area

As a user selects Certification Area from submenu of Texas, the user gets a list of certification area, which are divided among different grade levels. Figure 5.13 shows the grade levels and certification areas of Core Subjects. It also gives grade lists and a certification area list for World Languages.

![Figure 5.13. Texas – Certification area.](image)

As a user selects any certification area, it shows a program list associated with the respective certification area and grade. It also provides exam details for that particular certification area. In exam detail it provides: exam name, exam code, organization that administered particular exam and description and contact detail of that organization.
5.4.6 Texas — Region

As a user selects Region from submenu of Texas, the user gets region list with Texas region clickable map. Figure 5.14 shows the clickable map of Texas region. Users can click on any region in map to get list of service centers of that region. As a user selects any region in next screen the user gets list of program types to select. Program types are:

- University Based Programs
- Post-Baccalaureate Programs
- Alternative/Accelerated Programs

![Figure 5.14. Texas – Region.](image)

Then a user selects any type of program and next screen shows the list of service centers for respective region and respective program type. Figure 5.15 shows the screen which shows list of service center for select region and program type.

5.4.7 California — Credential

On Home Page, when a user clicks on California State in the USA map, the user redirects to California home page. Figure 5.16 shows the main screen of California State. This screen shows there are two options in California for credentialing:
5.4.8 California — Subject

As a user selects single subject credential in California, the user redirects to next screen of subject list. Figure 5.17 shows screen for single subject credential.
Figure 5.17. California – Subjects.

This screen shows submenu for single credential, which includes:

- Subjects
- Examination
- Requirements

Figure 5.17 shows the subject list for World Languages in California. As a user selects any subject it shows description about that subject and list of exams required getting credential in that subject. It also provides detail of Exam and contact information of organization that administered these exams.

5.4.9 California — Examination

As a user selects Examination from submenu of California, the user gets submenu of Examination, which includes type of exams.

- CSET
- SSAT
- Specialized
As a user selects any exam type the user gets list of subject for World Languages and Core Subjects. Selecting any subject gives details of the respective exam for the respective subject. This detail includes Exam Number, Exam Name, Beginning Date, Minimum Score, Passing Score, and Overall Passing Score. (See Figure 5.18.)

![Figure 5.18. California – Examination.](image)

**5.4.10 California — Requirements**

As a user selects Requirements from submenu of California, the user gets list of categories as shown in Figure 5.19. Categories are:

- Individual Prepared Outside of United States
- Individual Prepared Outside of Out of State
- Prepared in California
- Peace Corps Teaching Experience
- University Internship Credential

Each category has different set of requirements to get credentialing. This user interface gives clear idea to user for getting requirement of category in which he belongs.
As a user selects any category, the user redirects to next screen. As shown in Figure 5.20 requirements are divided into two sections:

- Five Year Credentialing
- Clear Credentialing

Each section again has two routes, depending on route selection user gets list of requirements. Two Routes are:

- More than two years experience
- Less than two years experience

### 5.5 System Security

To implement any system we need to protect it from attacks. For that we have to secure system using different methods. This section explains security methods used in this system.
5.5.1 Cryptography

Cryptography is a combination of encryption and decryption, which is a technique used for secure communication. There are different algorithms and techniques available for cryptography. In cryptography to encrypt any message a key gets use, which converts message into some code and another person cannot understand what that code is. To decrypt that code, it requires that one use the same key which has been used in encryption, and then a person can get original message.

Different cryptography algorithms:

- Hash function
- ISMACryp
- Ciphertext algorithm
- Symmetric-key algorithm, etc.

In this thesis a hash function has been used for encryption of password. As user fills out details on the registration page along with password and clicks on register, a controller class fetches user detail, which has been filled out the registration form and inserts all detail
in database to create new user account. During this process PHP controller first encrypt
password before inserting it in database.

Code:

```php
$pwd = $_GET['pwd_txt'];
$encrypted_pwd = md5($pwd);
/signupdata = new SignUpBean();
.signupdata->setPassword($encrypted_pwd);
```

Above code is for encryption used in this thesis. This thesis uses PHP’s inbuilt
function MD5 to encrypt password. MD5 function uses hash algorithm for encryption. This
function calculates the MD5 hash of $pwd using the RSA Data Security, MD5 Message-
Digest Algorithm, and returns that hash.

5.5.2 OAuth

OAuth provides a method for users to grant third-party access to their resources
without sharing their passwords. It also provides a way to grant limited access (in scope,
duration, etc.).

For example, a web user (resource owner) can grant a printing service (client) access
to her private photos stored at a photo sharing service (server), without sharing her username
and password with the printing service. Instead, she authenticates directly with the photo
sharing service, which issues the printing service delegation-specific credentials.

5.5.3 Beyond Client-Server

In the traditional client-server authentication model, the client uses its credentials to
access its resources hosted by the server. OAuth introduces a third role to this model: the
resource owner. In the OAuth model, the client (which is not the resource owner, but is
acting on its behalf) requests access to resources controlled by the resource owner, but hosted
by the server.

In order for the client to access resources, it first has to obtain permission from the
resource owner. This permission is expressed in the form of a token and matching shared-
secret. The purpose of the token is to make it unnecessary for the resource owner to share its
credentials with the client. Unlike the resource owner credentials, tokens can be issued with a restricted scope and limited lifetime, and revoked independently. [27]

In this thesis Facebook API has been used to implement “Login with Facebook” feature. In this feature user enters his/her login credentials of Facebook in Facebook login popup. When user enters Facebook login credentials on this website, it should be secure enough for user. As user provides Facebook login details, user directly get connects to Facebook server and his/her login credential gets authenticate there and third party (in this scenario Teacher Credentialing website) should not be able to fetch user’s Facebook login detail. OAuth in Facebook API has implemented this security, and OAuth takes care of user authentication.

5.5.4 OpenID

OpenID is an open standard that describes how users can be authenticated in a decentralized manner, eliminating the need for services to provide their own ad hoc systems and allowing users to consolidate their digital identities. Users may create accounts with their preferred OpenID identity providers and then use those accounts as the basis for signing on to any website which accepts OpenID authentication. The OpenID standard provides a framework for the communication that must take place between the identity provider and the OpenID acceptor (the ‘relying party’). An extension to the standard (the OpenID Attribute Exchange) facilitates the transfer of user attributes, such as name and gender, from the OpenID identity provider to the relying party (each relying party may request a different set of attributes, depending on its requirements) [28].

In this thesis Google API has been used to implement “Login with Google” feature. In this feature user enters his/her login credentials of Google in Google login page. When user enters Google login credentials on this website, then it should be secure enough then his/her Google login details are directly connecting to Google server and gets authenticate there and third party (in this scenario Teacher Credentialing website) should not be able to fetch user’s Facebook login detail. OpenID in Google API has implemented this security, and OpenID takes care of user authentication.
CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The thesis mainly focuses on user interface in a teacher credentialing website. Primary focus is on means to present teacher credentialing data whilst allowing a user to easily use the site and get helpful data. This thesis is also taking in consideration the accessibility feature for the non-technical user. A non-technical user can easily browse through the website for various teacher credentialing requirements. The main obstacles were to understand the teacher credentialing of Texas and design user friendly website using that data. This website also received appreciation from the members of the National Advisory Board during their Language Acquisition Resource Center site conference.

6.2 FUTURE ENHANCEMENT

- Design interface to insert/edit data into database for admin.
- Design and Implement forum feature for authorized user.
- Design and Implement interface for other states, which provide teacher-credentialing data for respective state.
- More features can be added, such as live chat with teacher, video calling.
- Design new interface for mobile devices like smart phones, tablets, iPad, etc.
- More social networking features can be add, such as +1 (Google+), Twitter, etc.
REFERENCES


