A REPORT ON

**FINDING AND FIXING VULNEREBALELIES IN WEB BASED APPLICATION i.e.**

**FACULTY’S APPOINTMENT SCHEDULER**

**Prepared By:**

Abhijeet Bajpai(2016btechcse101)

Devansh Aggarwal(2016btechcse108)

Rishabh Jain(2016btechcse120)

**Faculty Guide:-** Mr. Abhishek Vyas (Lecturer)



**Department of Computer Science & Engineering Institute of Engineering and Technology (IET) JK Lakshmipat University Jaipur**

**April, 2019**

**FINDING AND FIXING VULNEREBALELIES IN WEB BASED APPLICATION i.e.**

**STUDENT-FACULTY APPOINTMENT SCHEDULER**

Submitted in partial fulfillment of the requirements for the degree of

**Bachelor of Technology in Computer Science & Engineering**

**Prepared By:**

Abhijeet Bajpai(2016btechcse101)

Devansh Aggarwal(2016btechcse108)

Rishabh Jain(2016btechcse120)

**Faculty Guide:-** Mr. Abhishek Vyas (Lecturer)

**Department of Computer Science & Engineering Institute of Engineering and Technology (IET) JK Lakshmipat University Jaipur**

**April, 2019**

**CERTIFICATE**

This is to certify that the Major Project work entitled **“FINDING AND FIXING VULNEREBALELIES IN WEB BASED APPLICATION i.e. FACULTY’S APPOINTMENT SCHEDULER”** submitted by **Abhijeet Bajpai(2016BtechCSE101), Devansh Aggarwal (2016BtechCSE108), Rishabh Jain (2016Btech CSE128)**, towards the partial fulfilment of the requirements for the degree of **Bachelor of Technology in Computer Science & Engineering** of JK Lakshmipat University Jaipur is the record of work carried out by them under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for Major Project.

----------------------------------------

Mr. Abhishek Vyash Lecturer,

Department of Computer Science & Engineering

JK Lakshmipat University

Date of Submission:

## ACKNOWLEDGEMENTS

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We would like to express our warm gratitude to Mr. Abhishek Vyas, Lecturer, Dept. of Computer Science & Engineering, JK Lakshmipat University Jaipur for his valuable time and guidance that made the project work a success. We would like to express our heartfelt gratitude to **Prof Dr. Sanjay Goel**, **Director, Institute of Engineering & Technology**, **JK Lakshmipat University Jaipur** for his precious time, support and constant motivation in making our project successful. We would also like to express our gratitude towards member of **JK Lakshmipat University** for their kind co-operation and encouragement which help us in completion of this project.

We express our warm thanks & deep sense of gratitude to the individuals for their generous help in discussing the project and giving their valuable time in successful completion of this project. Time to time we got constructive suggestions, guidance and encouragement, which gave a constructive contribution to the completion of our project.

## LIST OF FIGURE

[Figure 1:-Use Case Diagram 12](file:///C:\Users\HP\Desktop\Application_security.docx#_Toc7756835)

[Figure 2:-Context Diagram 13](file:///C:\Users\HP\Desktop\Application_security.docx#_Toc7756836)

[Figure 3:-Faculty Activity Diagram 15](file:///C:\Users\HP\Desktop\Application_security.docx#_Toc7756837)

[Figure 4:-Student Activity 16](#_Toc7756838)

[Figure 5:-VEGA scan report 17](#_Toc7756839)

[Figure 6:-ZAP Scan Report 18](file:///C:\Users\HP\Desktop\Application_security.docx#_Toc7756840)

[Figure 7:-Password auto complete Vulnerability 19](#_Toc7756841)

[Figure 8:-Patched Code 20](#_Toc7756842)

[Figure 9:-Patched code for secure http only 20](#_Toc7756843)

[Figure 10:-Code for secure http secure flag 21](#_Toc7756844)

[Figure 11:-Server config 22](#_Toc7756845)

[Figure 12:- Cross site vulnerable Page 23](#_Toc7756846)

[Figure 13:- code to patch Cross site scripting 23](#_Toc7756847)

[Figure 14:-Page after Patching 24](#_Toc7756848)

Table of Contents

[ACKNOWLEDGEMENTS 4](#_Toc7757478)

[LIST OF FIGURE 5](#_Toc7757479)

[CHAPTER 1: INTRODUCTION 8](#_Toc7757480)

[**1.1 PROBLEM STATEMENT** 8](#_Toc7757481)

[**1.2 TECHNOLOGY USED** 8](#_Toc7757482)

[CHAPTER 2: REQUIREMENTS FOR PROPOSED WORK 10](#_Toc7757483)

[**2.1 PROGRAMMING LANGUAGE (SCRIPTING):-** 10](#_Toc7757484)

[**2.2 SOFTWARES:-** 10](#_Toc7757485)

[CHAPTER 3: REQUIREMENT ANALYSIS AND SPECIFICATION 11](#_Toc7757486)

[**3.1 REQUIREMENT ELICITATION:-** 11](#_Toc7757487)

[**3.2 REQUIREMENT ANALYSIS:-** 12](#_Toc7757488)

[**3.3 SYSTEM FEATURES** 13](#_Toc7757489)

[CHAPTER 4: FINDING VULNERABILITIES 17](#_Toc7757490)

[CHAPTER 5: PATCHING THE VULNEREBALITIES 19](#_Toc7757491)

[**1- Form Password Field with Auto complete Enabled** 19](#_Toc7757492)

[**2-Session cookie without Http Only Flag:-** 20](#_Toc7757497)

[**3- Session cookie Without Secure Flag:-** 21](#_Toc7757500)

[**4-Http Trace Support Detected:-** 21](#_Toc7757503)

[**5- Cross site Scripting (Persistent):-** 22](#_Toc7757506)

[REFERENCES 25](#_Toc7757509)

## CHAPTER 1: INTRODUCTION

### **1.1 PROBLEM STATEMENT**

A “Portal” is a web system which provides an interface for many people to communicate information by providing the functionalities and features to authenticate, storage , and evaluate and identify the users and provide them with an easy and customizable web interface for accessing information that are in interest of the user. In this project we come up with a similar teacher student portal which can be implemented in an educational organization for the purpose of saving time and checking the availability of the teacher to have an appointment with the student. The portal will provide an authentication platform for both side of users and feature to upload their status. These type of applications generally requires high security so to maintain the confidentiality, integrity and authenticity of the user. So, to remove the vulnerabilities in an application has been parameterized as one of the most important step of its development .The penetration testing exercise helps the developer to find out and patch the vulnerabilities in the application by scanning for the vulnerabilities and using one of many penetration tool available in market . This project will help an individual to gain knowledge on how to patch basic level vulnerabilities in an application and making it secure.

### **1.2 TECHNOLOGY USED**

To develop this Web utility, Hypertext Markup Language (honestly called HTML) at the side of Cascading Style Sheets (CSS), Hypertext Preprocessor (PHP) and SQL for database are used.

**Hypertext Markup Language (HTML):-** It is the usual markup language for developing internet pages and web programs. HTML factors are the constructing blocks of HTML pages. With HTML constructs, pictures and different items together with interactive forms may be embedded into the rendered page. HTML gives a way to create based files by means of denoting structural semantics for text along with headings, paragraphs, lists, hyperlinks and other things.

**Cascading Style Sheets (CSS):-** It is a Style sheet language used for describing the presentation of a record written in a markup language like HTML. CSS is a cornerstone era of the World Wide Web. CSS is designed to allow the separation of presentation and content material, such as format, shades, and fonts. This separation can enhance content material accessibility, offer greater flexibility and control inside the specification of presentation traits, enable multiple web pages to proportion formatting by means of specifying the applicable CSS in a separate .Css document, and decrease complexity and repetition within the structural content material.

**PHP Hypertext Pre-processor**(**PHP**):- Php is a [general-purpose programming](https://en.wikipedia.org/wiki/General-purpose_programming_language) [language](https://en.wikipedia.org/wiki/General-purpose_programming_language) originally designed for [web development.](https://en.wikipedia.org/wiki/Web_development) PHP code may be executed with a [command line](https://en.wikipedia.org/wiki/Command-line_interface) [interface](https://en.wikipedia.org/wiki/Command-line_interface) (CLI), embedded into [HTML](https://en.wikipedia.org/wiki/HTML) code, or it can be used in combination with various [web template systems,](https://en.wikipedia.org/wiki/Web_template_system) web content management systems, and [web frameworks.](https://en.wikipedia.org/wiki/Web_framework)

**MySQL** is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) under the terms of the [GNU General Public](https://en.wikipedia.org/wiki/GNU_General_Public_License) [License,](https://en.wikipedia.org/wiki/GNU_General_Public_License) and is also available under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) licenses. MySQL is a component of the [LAMP](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) [web application](https://en.wikipedia.org/wiki/Web_application) [software stack](https://en.wikipedia.org/wiki/Web_application) (and [others](https://en.wikipedia.org/wiki/List_of_AMP_packages)), which is an acronym for [*Linux,*](https://en.wikipedia.org/wiki/Linux)[*Apache,*](https://en.wikipedia.org/wiki/Apache_HTTP_Server) *MySQL,* [*Perl*](https://en.wikipedia.org/wiki/Perl)*/*[*PHP*](https://en.wikipedia.org/wiki/PHP)*/*[*Python*.](https://en.wikipedia.org/wiki/Python_(programming_language)) MySQL is used by many database-driven web applications, including [Drupal](https://en.wikipedia.org/wiki/Drupal), [Joomla](https://en.wikipedia.org/wiki/Joomla), [phpBB,](https://en.wikipedia.org/wiki/PhpBB) and [WordPress.](https://en.wikipedia.org/wiki/WordPress)

Based on UML methodology, this application design includes Data Flow Diagrams, E- R Diagrams, Use Case Diagrams. These diagrams are useful to visualize the architecture of a system.

**Data flow diagram** (DFD):- DFD is a way of representing a flow of a data of a [process](https://en.wikipedia.org/wiki/Process) or a system (usually an [information system](https://en.wikipedia.org/wiki/Information_system)) The DFD also provides information about the outputs and inputs of each entity and the process itself. A data flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a [flowchart.](https://en.wikipedia.org/wiki/Flowchart)

**Use case diagram: -** It is useful to identify different functions and show how the actor users can interact with these functions based on their roles. An actor can be a donor, requester, admin based on the actor’s role using the application. The Primary actor is the requester user who uses the application to notify the Donors. The use cases can be Blood Request feed, Book appointment, invite friends, donation history etc.

## CHAPTER 2: REQUIREMENTS FOR PROPOSED WORK

#### 

### **2.1 PROGRAMMING LANGUAGE (SCRIPTING):-**

The following are the languages which were required to build a website:

* + - Hyper Text Markup Language (HTML),
    - Cascading Style Sheets (CSS),
    - Hypertext Preprocessor (PHP),
    - Bootstrap,
    - JavaScript,
    - My Structured Query Language (MySQL).

### **2.2 SOFTWARES:-**

Sublime Text, Web Browser And XAMPP Server(Server+PhpMyAdmin).

## CHAPTER 3: REQUIREMENT ANALYSIS AND SPECIFICATION

The Main aim of this Project is to elicit user behavior about the faculty appointment management and to apprehend and describe styles of these ideals. This mixed approach became carried out in two components. The first is requirement elicitation that is amassing of required necessities for the System with the assist of users . The different one is requirement analysis wherein necessities are analyzed a good way to identify inconsistencies or defects within the device and additionally resolves them.

### **3.1 REQUIREMENT ELICITATION:-**

The primary approach used in this Faculty appointment System for collecting requirement is the Use Case Diagram approach. It is the graphical representation used for the description of user requirements.

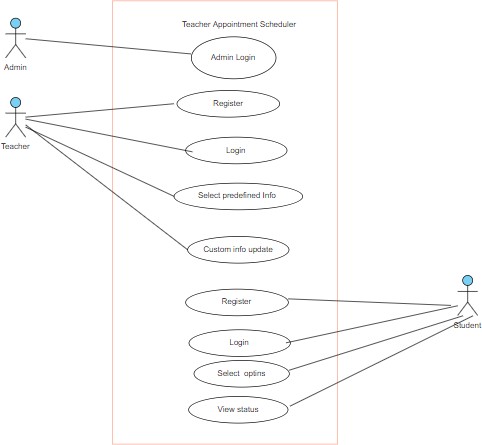


Figure :-Use Case Diagram

### **3.2 REQUIREMENT ANALYSIS:-**

Requirement evaluation lets in to perceive the patterns, subject matters and categories emerging from consumer’s angle. It will assist to understand various viewpoints and discovers the all initial requirements and subsequently produce the required provider for the person.

The number one technique used on this Faculty appointment System for evaluation and modeling of requirement is thru the Data Flow Diagram or the Context Diagram (Level 0 DFD). It is the graphical illustration which defines the bounds and interface of the proposed system.

The Context Diagram for Blood Source Management System is given under:

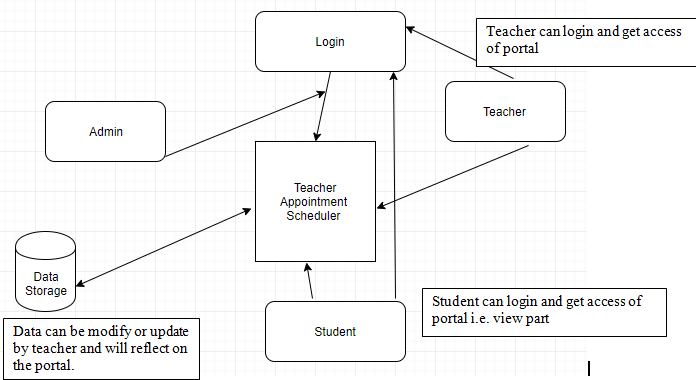


Figure :-Context Diagram

### **3.3 SYSTEM FEATURES**

#### Faculty Member Activity:-

**Login page:** Provide a login page to the faculty so that they can login with their specific login details and then redirect to their respective portal.

**Main activity:** The user or faculty shall be able to update their status.

**Logout:** The user shall be able to Logout. Stimulus/Response Sequences:-

#### Login Page and main activity for Faculty Member

Stimulus: User clicks on Login Link. Response: Login Page is displayed.

Stimulus: User Enters Username and Password.

Response: Username and Password are validated from MySQL Database. Stimulus: User Clicks on Login Button.

Response: Home Page is displayed if Username and Password is correct else Error Message is displayed*.*

Stimulus: User will update Information from the home page or specific portal.

Response: The information will update on the database and that would be the current status of the user which will reflect on the student’s page.

#### Logout

Stimulus: User clicks on Logout link.

Response: User is logged out and login page is displayed.

Figure :-Faculty Activity Diagram

Logout

Updated information

Home page (Portal)

Login Page

#### Student Activity:-

**Login page:** Provide a login page to the faculty so that they can login with their specific login details and then redirect to their respective portal.

**Main activity:** The user or faculty shall be able to update their status.

**Logout:** The user shall be able to Logout.

1. **Login Page and main activity for** StudentStimulus: User clicks on Login Link. Response: Login Page is displayed.

Stimulus: User Enters Username and Password.

Response: Username and Password are validated from MySQL Database. Stimulus: User Clicks on Login Button.

Response: Home Page is displayed if Username and Password is correct else Error Message is displayed*.*

#### Logout

Stimulus: User clicks on Logout link.

Response: User is logged out and login page is displayed.

Logout

View portal

Login for student

Figure :-Student Activity

## CHAPTER 4: FINDING VULNERABILITIES

To find the vulnerabilities we have used three tools namely, Vega and OWASP- ZAP.



Figure :-VEGA scan report

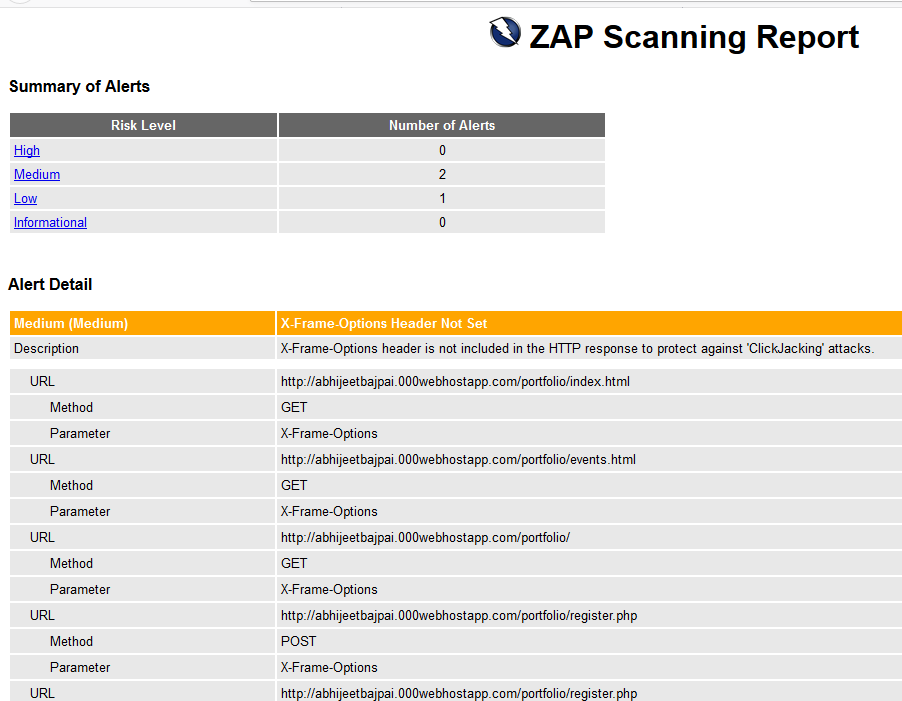




Figure :-ZAP Scan Report

## CHAPTER 5: PATCHING THE VULNEREBALITIES

### **1- Form Password Field with Auto complete Enabled**

**Impact**

* A password value may be stored on the local filesystem of the client.
* Locally stored passwords could be retrieved by other users or malicious code.



Figure 7:-Password auto complete Vulnerability

### Fig- 7 is the representation of Form Password Field with Auto complete Enabled, whenever user will suppose to enter password, input field reflect the entire previous entered password.

**Remediation**

* The form declaration should have an autocomplete attribute with its value set to "off".

### **C:\Users\HP\Pictures\Screenshots\Screenshot (229).png**

Figure :-Patched Code

### Session Cookie Wit

### **2-Session cookie without Http Only Flag:-**

**Discussion**

Vega has detected that a session cookie may have been set without the HttpOnly flag. When this flag is not present, it is possible to access the cookie via client-side script code. The HttpOnly flag is a security measure that can help mitigate the risk of cross-site scripting attacks that target session cookies of the victim. If the HttpOnly flag is set and the browser supports this feature, attacker-supplied script code will not be able to access the cookie.

**Remediation**

* When creating the cookie in the code, set the HttpOnly flag to true.

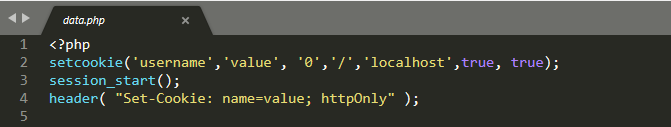


Figure :-Patched code for secure http only

### **3- Session cookie Without Secure Flag:-**

**Impact**

* Cookies can be exposed to network eavesdroppers.
* Session cookies are authentication credentials; attackers who obtain them can get unauthorized access to affected web applications.

**Remediation**

* When creating the cookie in the code, set the secure flag to true.

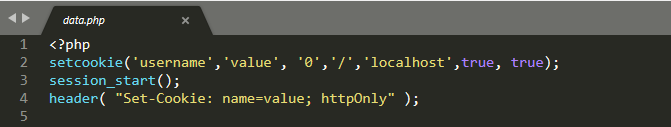


Figure :-Code for secure http secure flag

### **4-Http Trace Support Detected:-**

**Impact**

* Allowing HTTP TRACE can permit cross-site tracing.
* Attackers may be able to use cross-site tracing with cross-site scripting retrieve the value of HttpOnly cookies.

**Remediation**

* For Apache based servers, the TraceEnable directive can be used to disable support for HTTP TRACE.
* For IIS based servers, the EnableTraceMethod registry setting controls support for HTTP TRACE.

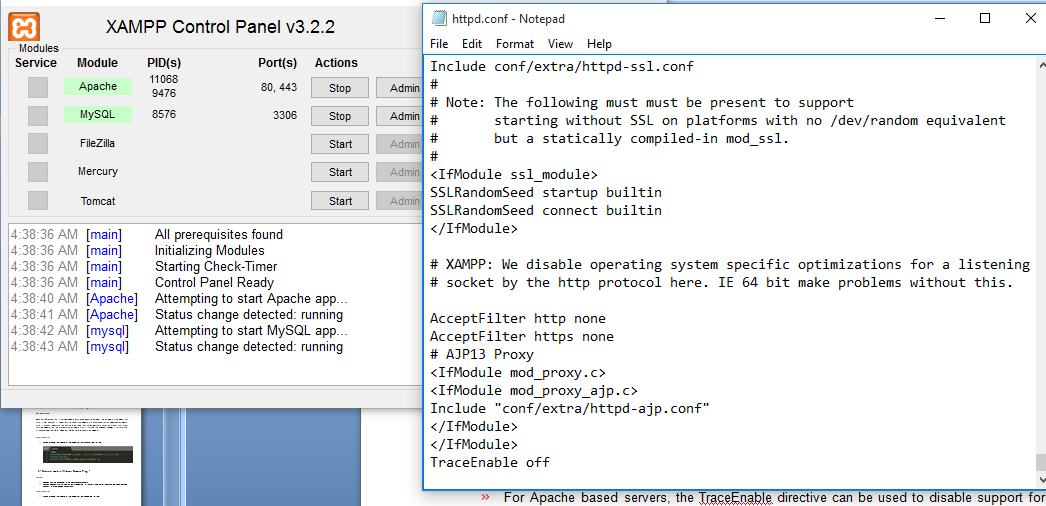


Figure :-Server config

### **5- Cross site Scripting (Persistent):-**

### This Vulnerability found by hit and trial method, by inserting some malicious (script tags) code. it found that web application has persistent XSS vulnerable.

### Here is brief about the impact of Stored or persistent XSS.

Stored XSS (also known as persistent or second-order XSS) arises when an application receives data from a UN trusted source and includes that data within its later HTTP responses in an unsafe way.

The data in question might be submitted to the application via HTTP requests; for example, comments on a blog post, user nicknames in a chat room, or contact details on a customer order. In other cases, the data might arrive from other un trusted sources; for example, a webmail application displaying messages received over SMTP, a marketing application displaying social media posts, or a network monitoring application displaying packet data from network traffic.

Here is a simple example of a stored XSS vulnerability. A message board application lets users submit messages, which are displayed to other users:

<p>Hello, this is my message!</p>

The application doesn't perform any other processing of the data, so an attacker can easily send a message that attacks other users:

<p><script>/\* Bad stuff here... \*/</script></p>

**Trial:- <img src="does-not-exist" onerror="alert(document.cookie)">**

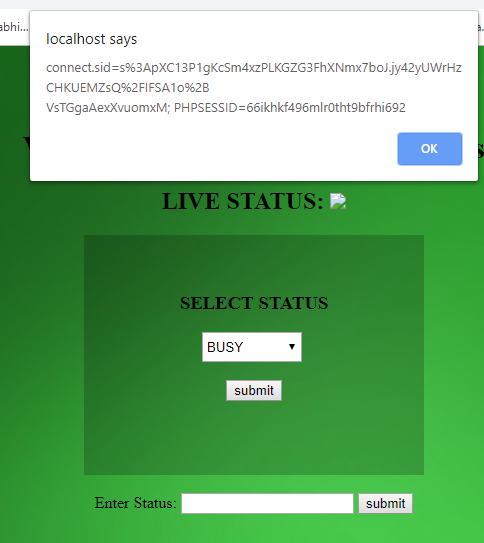
****

Figure :- Cross site vulnerable Page

Remediation:-

**htmlspecialchar() Function:-**

The htmlspecialchars() function converts some predefined characters to HTML entities.

The predefined characters are:

* & (ampersand) becomes &amp;
* " (double quote) becomes &quot;
* ' (single quote) becomes &#039;
* < (less than) becomes &lt;
* > (greater than) becomes &gt;

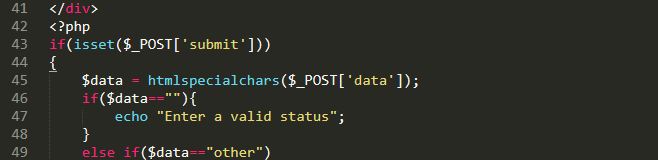


Figure :- code to patch Cross site scripting

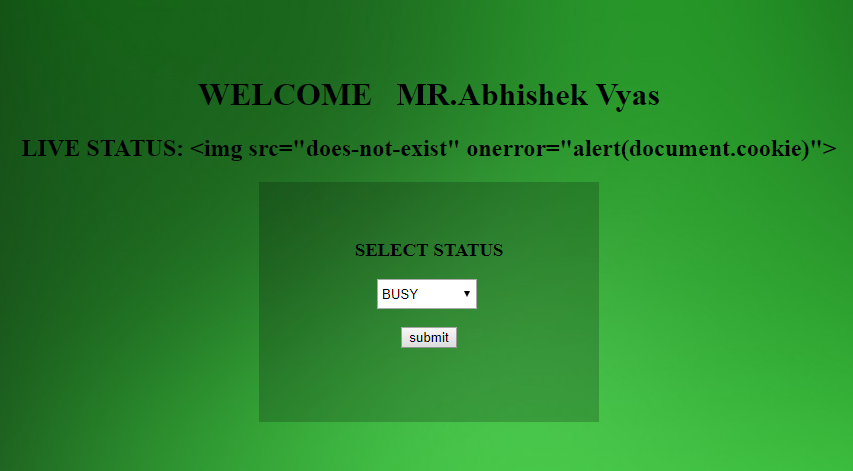


Figure :-Page after Patching

## REFERENCES

1-Hire PHP Developer, PHP Developer, PHP Development Company, PHP Application Development, Hire PHP Programmer, PHP Developer. (2012). *SciVee*. doi:10.4016/47862.01

2-Fox, D. (2012). Cross Site Scripting (XSS). *Datenschutz Und Datensicherheit - Dud*, *36*(11), 840-840. doi: 10.1007/s11623-012-0284-2

3- https://www.owasp.org/index.php/HttpOnly

4-[https://portswigger.net/kb/issues/00500200\_ssl-cookie-without-secure-flag- set](https://portswigger.net/kb/issues/00500200_ssl-cookie-without-secure-flag-%20%20%20set)