A PROJECT REPORT ON
Online Fee Payment

A Mini Project Report Submitted to Jawaharlal Nehru Technological College, Kakinada

BACHELOR OF

TECHNOLOGY IN

COMPUTER SCIENCE AND ENGINEERING (AI & ML)

Submitted By

Mr. P. Leela Prasad (20KT1A4238)

Under the Esteemed Guidance of
Mrs.P. Sri Silpa, M.Tech, (Ph.D.)

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

POTTI SRIRAMULU CHALAVADI MALLIKARJUNA RAO COLLEGE OF ENGINEERING & TECHNOLOGY
(Approved by AICTE New Delhi, Affiliated to JNTU-Kakinada) KOTHAPET, VIJAYAWADA-520001, A.P
2020-2024

POTTI SRIRAMULU CHALAVADI MALLIKHARJUNARAO

COLLEGE OF ENGINEERING & TECHNOLOGY

KOTHAPET, VIJAYAWADA-520001.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(AI & ML)

CERTIFICATE

This is to certify that the project work entitled **"ONLINE FEE PAYMENT"** is a bonafide work carried out by **Mr. P. Leela Prasad (20KT1A4238),** Fulfillment for the award of the degree of Bachelor of Technology in **COMPUTER SCIENCE AND ENGINEERING (AI & ML)** of Jawaharlal Nehru Technological College, Kakinada during the year **2021-2022.** It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the above degree.

Project Guide **Head of the Department**

External Examiner

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

We owe a great many thanks to a great many people who helped and supported and suggested us in every step.We are glad for having the support of our principal **Dr. K. Sri Rama Krishna** who inspired us with his words filled with dedication and discipline towards work.We express our gratitude towards **Dr**. **A. Pathanjali Sastri, Professor & HoD of CSE** for extending his support through training classes which had been the major source to carry out our project.We are very much thankful to **Mrs. P.Sri Silpa**, **M. Tech, (Ph.D.)** Associate Professor/ Professor, Guide of our project for guiding and correcting various documents of ours with attention and care. He/ She has taken pain to go through the project and make necessary corrections as and when needed.Finally, we thank one and all who directly and indirectly helped us to complete our project successfully.

Project Associates

P. Leela Prasad (20KT1A4238)

DECLARATION

DECLARATION

This is to declare that the project entitled **"ONLINE FEE PAYMENT"** submitted by us in the partial fulfillment of requirements for the award of the degree of **Bachelor of Technology in Computer Science & Engineering** in **Potti Sriramulu Chalavadi Mallikarjuna Rao College of Engineering and Technology**, is bonafide record of project work carried out by us under the supervision and guidance of **Ms. P.Sri Silpa, Associate Professor of CSE**. As per our knowledge, the work has not been submitted to any other institute or universities for any other degree.

Project Associates

P. Leela Prasad (20KT1A4238)

ABSTRACT

ABSTRACT

PSCMRCET has a large number of students who pay all the college fees through cash deposits, electronic funds transfer or bank drafts to the college's accounts in specific bank branches.These methods of paying fees have not been efficient enough especially during periods of tests and examinations when most of the students are paying fees to meet the requirements for entering examination rooms.The process of paying fees is characterized by long queues, too much waiting by students and congestion at banks where payments are made.This has always resulted in students missing to sit for their tests and/or examinations while they are queuing to Make payments.It was upon such background that the researchers embarked on the project to develop of an alternative system that enables online fees payment by students and their Guidance. With the use of questionnaires, interviews, observation and document reviews, data was collected from project stakeholders and analyzed. Data flow diagrams and Entity relationship diagrams were used to accomplish system analysis and design. The system was implemented using Apache web server, MySQL database server, Hypertext Preprocessor, Hypertext markup language, Cascading style sheets and JavaScript. System testing and validation was also done by allowing users of the system interact with it using test data. Findings showed that most of the students were unsatisfied with the current modes of paying fees to the college and agree that an online fees payment system can improve the process of fees payment. The result of the project was an online fees payment system for Potti Sriramulu Chalavadi Mallikarjunarao College of Engineering & Technology (PSCMRCET) and researchers recommend the college to implement the system that provides relief of the long endured problems of the current modes of payment at the college.

CONTENTS

CONTENTS

LIST OF FIGURES

LIST OF FIGURES

List of Tables

INTRODUCTION

Chapter 1

Introduction

1.1 Hypertext Markup Language:

HTML, in full **hypertext markup language**, a formatting system for displaying material retrieved over the Internet. Each retrieval unit is known as a Web page (from World Wide Web), and such pages frequently contain hypertext links that allow related pages to be retrieved. HTML is the markup language for encoding Web pages. It was designed by the British scientist Sir Tim Berners-Lee at the CERN nuclear physics laboratory in Switzerland during the 1980s. HTML markup tags specify document elements such as headings, paragraphs, and tables. They mark up a document for display by a computer program known as a Web browser. The browser interprets the tags, displaying the headings, paragraphs, and tables in a layout that is adapted to the screen size and fonts available to it.

HTML documents also contain anchors, which are tags that specify links to other Web pages. An anchor has the form <A HREF= “http://www.britannica.com”> Encyclopedia Britannica</A>, where the quoted string is the URL (universal resource locator) to which the link points (the Web “address”) and the text following it is what appears in a Web browser, underlined to show that it is a link to another page. What is displayed as a single page may also be formed from multiple URLs, some containing text and others graphics.

1.1 Source code for sample html

```
<!DOCTYPEhtml>
<html>
<body>
<h1>MyFirstHeading</h1>
<p>Myfirstparagraph.</p>
</body>
</html>
```

1.1 Result

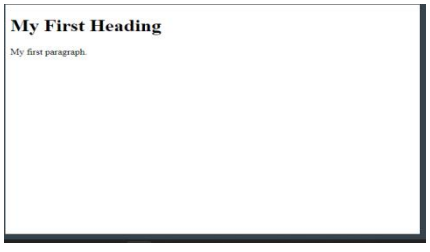

Figure: 1.1 : sample

1.2 Cascading Style Sheet

Cascading Style Sheets (CSS) is a powerful tool that transforms the presentation of a document or a collection of documents, and it has spread to nearly every corner of the web and into many ostensibly non-web environments. For example, Gecko-based browsers use CSS to affect the presentation of the browser chrome itself, many RSS clients let you apply CSS to feeds and feed entries, and some instant message clients use CSS to format chat windows. Aspects of CSS can be found in the syntax used by JavaScript frameworks, and even in JavaScript itself. It's everywhere!

CSS was first proposed in 1994, just as the web was beginning to really catch on. At the time, browsers gave all sorts of styling power to the user—the presentation preferences in Mosaic, for example, permitted all manner of font family, size, and color to be defined by the user on a per-element basis. None of this was available to document authors; all they could do was mark a piece of content as a paragraph, as a heading of some level, as preformatted text, or one of a handful of other element types. If a user configured his browser to make all level-one headings tiny and pink and all level-six headings huge and red, well, that was his lookout.

It was into this milieu that CSS was introduced. Its goal was to provide a simple, declarative styling language that was flexible for authors and, most importantly, provided styling power to authors and users alike. By means of the "cascade," these styles could be combined and prioritized so that both authors and readers had a say—though readers always had the last say.

Work quickly advanced, and by late 1996, CSS1 was finished. While the newly established CSS Working Group moved forward with CSS2, browsers struggled to implement CSS1 in an interoperable way. Although each piece of CSS was fairly simple on its own, the combination of those pieces created some surprisingly complex behaviors. There were also some unfortunate missteps in early implementations, such as the infamous discrepancy in box model implementations. These problems threatened to derail CSS altogether, but fortunately some clever proposals were implemented, and browsers began to harmonize. Within a few years, thanks to increasing interoperability and high-profile developments such as the CSS-based redesign of *Wired* magazine and the CSS Zen Garden, CSS began to catch on.

Before all that happened, though, the CSS Working Group had finalized the CSS2 specification in early 1998. Once CSS2 was finished, work immediately began on CSS3, as well as a clarified version of CSS2 called CSS2.1. In keeping with the spirit of the times, CSS3 was constructed as a series of (theoretically) standalone modules instead of a single monolithic specification. This approach reflected the then-active XHTML specification, which was split into modules for similar reasons.

The rationale for modularizing CSS3 was that each module could be worked on at its own pace, and particularly critical (or popular) modules could be advanced along the W3C's progress track without being held up by others. Indeed, this has turned out to be the case. By early 2012, three CSS3 modules (along with CSS1 and CSS 2.1) had reached full Recommendation status—CSS Color Level 3, CSS Namespaces, and Selectors Level 3. At that same time, seven modules were at Candidate Recommendation status, and several dozen others were in various stages of Working Draft-ness. Under the old approach, colors, selectors, and namespaces would have had to wait for every other part of the specification to be done or cut before they could be part of a completed specification. Thanks to

modularization, they didn't have to wait.

The flip side of that advantage is that it's hard to speak of a single "CSS3 specification." There isn't any such thing, nor can there be. Even if every other CSS module had reached level 3 by, say, late 2016 (they didn't), there was already a Selectors Level 4 in process. Would we then speak of it as CSS4? What about all the "CSS3" features still coming into play? Or Grid Layout, which had not then even reached Level 1?

So while we can't really point to a single tome and say, "There is CSS3," we can talk of features by the module name under which they are introduced. The flexibility modules permit more than makes up for the semantic awkwardness they sometimes create. (If you want something approximating a single monolithic specification, the CSS Working Group publishes yearly "Snapshot" documents.)

With that established, we're almost ready to start understanding CSS. First though, we must go over markup.

1.3 JAVASCRIPT

JavaScript often abbreviated **JS**, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client side for web page behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions.It is multi-paradigm, supporting event-driven, functional, and imperative progra-mming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O. JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js. Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design. An event that can be handled is something happening in a browser window, including a document loading, the user clicking a mouse button, the user pressing a key, and the browser screen changing size. When a function is assigned to handle an event type, that function is run when an event of the event type occurs.

An event handler can be assigned in the following ways:

1. Via an element attribute directly in HTML: <body onload="alert('Hello World!');">
2. Via JavaScript, by assigning the event type to an element attribute: document.onclick = clickHandler;
3. Via JavaScript by a direct call to the addEventListener() method of an element.

A handler that is assigned from a script uses the syntax '[element].[event] = [function];', where [element] is a page element, [event] is the name of the selected event and [function] is the name of the function that is called when the event takes place.

For example:

```
document.onclick = clickHandler;
```

This handler will cause the function clickHandler() to be executed whenever the user clicks the mouse anywhere on the screen. Note that when an event handler is assigned, the function name does not end with parentheses. We are just pointing the event to the name of the function. The clickHandler() function is defined like this:

```
function clickHandler(event) {
  //some code here
}
```

In some browsers the event must be explicitly passed to the function, so as a precaution it's often best to include a conditional to test that the event variable has been passed, and if it hasn't then to use an alternative method that works on those other browsers:

```
function clickHandler(event) {
  event = event || window.event;
  //some code here
}
```

Elements within a document can also be assigned event handlers. For example:

```
document.getElementsByTagName('a')[0].onclick = linkHandler;
```

This will cause the linkHandler() function to be executed when the user clicks the first link on the page. Keep in mind that this style of handler assignment depends on the link's position inside the page. If another link tag is added before this one, it will take over the handler from the original link. A best practice is to maintain the separation of code and page structure by assigning each link an identifier by using the id attribute.

```
<a id="faqLink" href="faq.html">Faq</a>
```

A handler assignment can then work regardless of where the element is positioned.

```
document.getElementById('faqLink').onclick = linkHandler;
```

Events are actions that can be detected by JavaScript, and the event object gives information about the event that has occurred. Sometimes we want to execute a JavaScript when an event occurs, such as when a user clicks a button. Events are normally used in combination with functions, and the function will not be executed before the event occurs! JavaScript event

handlers are divided into two types:

1. Interactive event handlers - depend on user interaction with the HTML page; ex. clicking a button
2. Non-Interactive event handlers - do not need user interaction; ex. on load

1.4 PHP

PHP is a general-purpose scripting language geared toward web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994.The PHP reference implementation is now produced by The PHP Group. PHP originally stood for ***Personal Home Page***,but it now stands for the recursive initialism ***PHP: Hypertext Preprocessor***.

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside the web context, such as standalone graphical applications and robotic drone control. PHP code can also be directly executed from the command line. The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on a variety of operating systems and platforms. The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the *de facto* standard which other implementations aimed to follow. Since 2014, work has gone on to create a formal PHP specification. W3Techs reports that, as of January 2022, "PHP is used by 78.1% of all the websites whose server-side programming language we know." PHP version 7.4 is the most used version. Support for version 7.3 was dropped on 6 December 2021.

PHP functions

s.no	Function name	description
1	Connection to localhost	$servername = "localhost";
2	Selecting database	$conn=new mysqli($servername,$username, $password);
3	Mysqli conn	Mysqli query(connection, query,resultmode)
4	fetch array	$mysqli_result -> fetch_array(resultttype)
5	Mysqli close	$mysqli -> close();

1.5 MySQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter,and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for *Linux, Apache, MySQL, Perl/PHP/Python*. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube.

MySQL Commands

1. SELECT — extracts data from a database

   ```
   SELECT column_name
   FROM table_name;
   ```

   SELECT statements fetch data from a database.

2. UPDATE — updates data in a database

   ```
   UPDATE table_name
   SET some_column = some_value
   WHERE some_column = some_value;
   ```

   UPDATE statements allow us to edit rows in a table.

3. DELETE — deletes data from a database

   ```
   DELETE FROM table_name
   WHERE some_column = some_value;
   ```

   DELETE statements remove rows from a table.

4. INSERT INTO — inserts new data into a database

```
INSERT INTO table_name (column_1, column_2, column_3)
VALUES (value_1, 'value_2', value_3);
```

INSERT statements add a new row to a table.

5. CREATE DATABASE — creates a new database

   ```
   CREATE DATABASE databasename;
   ```

   CREATE DATABASE statements create a new SQL database.

6. ALTER DATABASE — modifies a database

   ```
   ALTER DATABASE database_name
   [COLLATE collation_name ]
   ```

   ALTER DATABASE statements change the characteristics of a database.

7. CREATE TABLE — creates a new table

   ```
   CREATE TABLE table_name (
   column_1 datatype,
   column_2 datatype,
   column_3 datatype
   );
   ```

   CREATE TABLE statements create a new table in the database.

8. ALTER TABLE — modifies a table

   ```
   ALTER TABLE table_name
   ADD column_name datatype;
   ```

   ALTER TABLE statements add, delete, or modify columns in an existing table.

9. DROP TABLE — deletes a table

   ```
   DROP TABLE table_name;
   ```

   DROP TABLE statements drop an existing table in a database.

10. CREATE INDEX — creates an index

    ```
    CREATE INDEX index_name
    ON table_name (column_name1, column_name2…);
    ```

    Index statements create on existing tables to retrieve the rows quickly.

SOFTWARE REQUIREMENTS

Chapter 2
SOFTWARE REQUIREMENTS

2.1 Visual studio Code

Visual Studio Code, also commonly referred to as **VS Code**,is a source-code editor made by Microsoft for Windows, Linux and macOS.Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 70% of 82,000 respondents reporting that they use it.

Features

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python, C++ and Fortran. It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

Out of the box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

Figure 2.1.1:Visual Studio Code Insiders logo

Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language. It supports many programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette.

Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, debuggers, time travel debuggers, perform static code analysis, and add code linters using the Language Server Protocol.

Source control is a built-in feature of Visual Studio Code. It has a dedicated tab inside of the menu bar where you can access version control settings and view changes made to the current project. To use the feature you must link Visual Studio Code to any supported version control system (Git, Apache Subversion, Perforce, etc.). This allows you to create repositories as well as to make push and pull requests directly from the Visual Studio Code program.

Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software. Visual Studio Code allows users to set the code page in which the active document is saved, the newline character, and the programming language of the active document. This allows it to be used on any platform, in any locale, and for any given programming language. Visual Studio Code collects usage data and sends it to Microsoft, although this can be disabled. Due to the open-source nature of the application, the telemetry code is accessible to the public, who can see exactly what is collected.

Installation of vs code

How to install Visual Studio code

Step 1 :

Download VS code from here (https://code.visualstudio.com/download).

Step 2 :

- Download the Visual Studio Code installer for Windows. Once it is downloaded, run the installer VSCodeUserSetup-{version}.exe). Then, run the file – it will only take a minute.

Accept the agreement and click “next.”

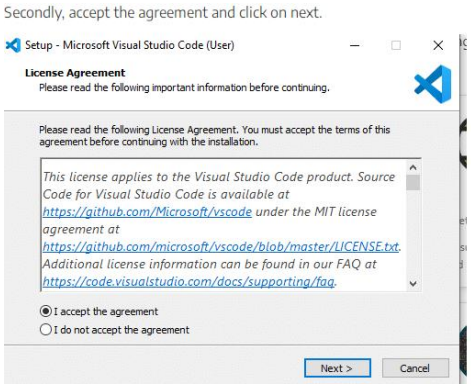

Figure 2.1.2:Accept the agreement

- After accepting all the requests press finish button. By default, VS Code installs under: **"C:\users{username}\AppData\Local\Programs\Microsoft VS Code."**

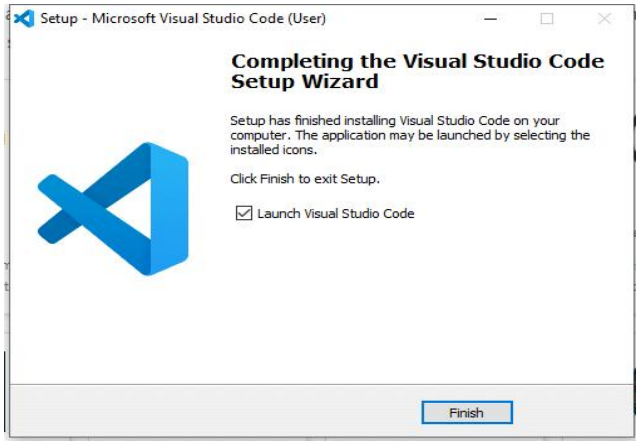

Figure 2.1.3:Finish The Setup

- If the installation is successful, you will see the following:

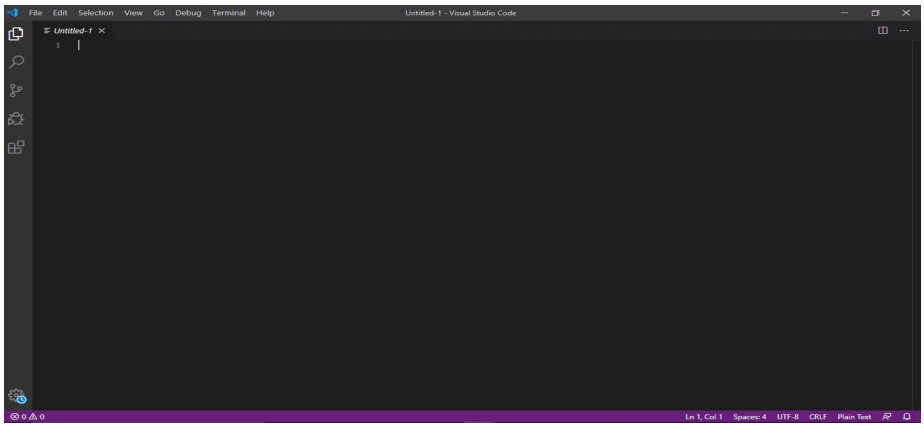

Figure 2.1.4: Main Page

2.2 XAMPP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

XAMPP's ease of deployment means a WAMP or LAMP stack can be installed quickly and simply on an operating system by a developer, with the advantage that common add-in applications such as WordPress and Joomla! can also be installed with similar ease using Bitnami.

Installing XAMPP

Our XAMPP tutorial will take you through the installation process for the software package on Windows. If you're using Linux or Mac OS X, then the steps listed below for the installation process may differ.

Step 1: Download

XAMPP is a release made available by the non-profit project Apache Friends. Versions with PHP 5.5, 5.6, or 7 are available for download on the Apache Friends website.

Step 2: Run .exe file

Once the software bundle has been downloaded, you can start the installation by double clicking on the file with the ending .exe.

Step 3: Deactivate any antivirus software

Since an active antivirus program can negatively affect the installation process, it's recommended to temporarily pause any antivirus software until all XAMPP components have successfully been installed.

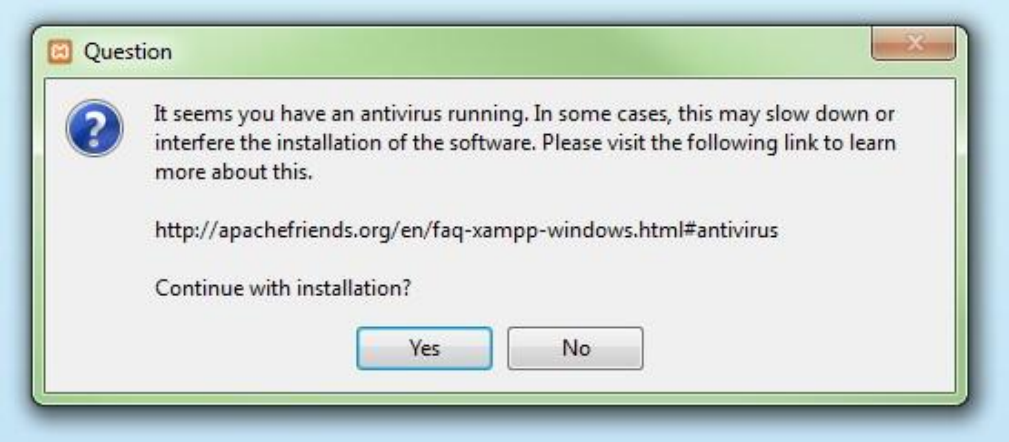

Figure 2.2.1:Conformation Of Installation

Before installing XAMPP, it is advisable to disable the anti-virus program temporarily

Step 4: Deactivate UAC

User Account Control (UAC) can interfere with the XAMPP installation because it limits writing access to the C: drive, so we recommend you deactivate this too for the duration of the installation process. To find out how to turn off your UAC, head to the Microsoft Windows support pages.

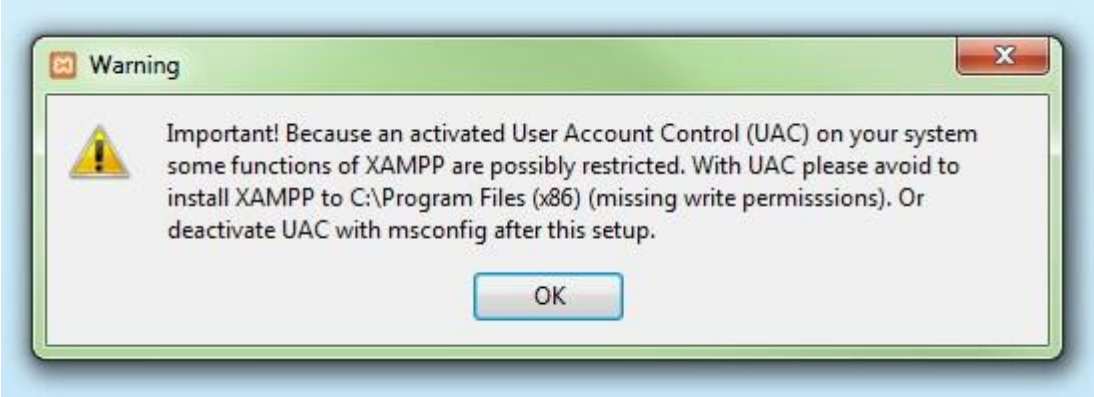

Figure 2.2.2: Image of Warning

User account control can affect the installation of XAMPP

Step 5: Start the setup wizard

After you've opened the .exe file (after deactivating your antivirus program(s) and taken note of the User Account Control, the start screen of the XAMPP setup wizard should appear automatically. Click on 'Next' to configure the installation settings.

Figure 2.2.3: Setup box

You can start the setup on the startup screen

Step 6: Choose software components

Under 'Select Components', you have the option to exclude individual components of the XAMPP software bundle from the installation. But for a full local test server, we recommend you install using the standard setup and all available components. After making your choice, click 'Next'.

Figure 2.2.4:Changing Directory

In the dialog window entitled 'select components', you can choose the software components before installation

Step 7: Choose the installation directory

In this next step, you have the chance to choose where you'd like the XAMPP software packet to be installed. If you opt for the standard setup, then a folder with the name XAMPP will be created under C:\ for you. After you've chosen a location, click 'Next'.

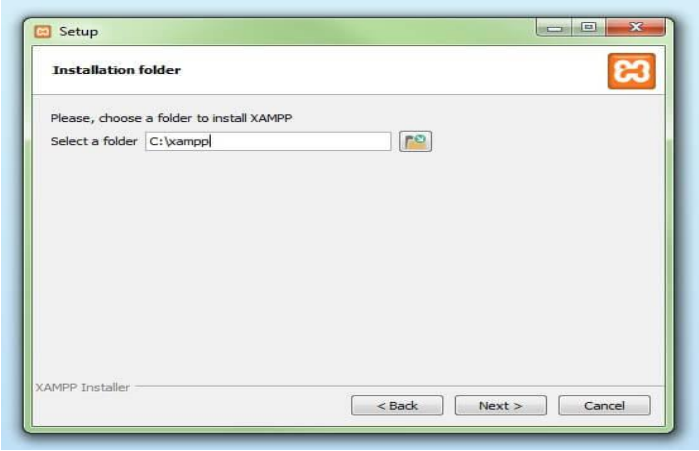

Figure 2.2.5:File Path

For the next step, you need to select the directory where XAMPP should be installed

Step 8: Start the installation process

Once all the aforementioned preferences have been decided, click to start the installation. The setup wizard will unpack and install the selected components and save them to the designated directory. This process can take several minutes in total. You can follow the progress of this installation by keeping an eye on the green loading bar in the middle of the screen.

Figure 2.2.6:Downloading Box

According to the default settings, the selected software components are unpacked and installed in the target folder

Step 9: Windows Firewall blocking

Your Firewall may interrupt the installation process to block the some components of the XAMPP. Use the corresponding check box to enable communication between the Apache server and your private network or work network. Remember that making your XAMPP server available for public networks isn't recommended.

Step 10: Complete installation

Once all the components are unpacked and installed, you can close the setup wizard by clicking on 'Finish'. Click to tick the corresponding check box and open the XAMPP Control Panel once the installation process is finished.

Figure 2.2.7:Figure of Finished Download

By clicking on 'finish', the XAMPP Setup Wizard is completed

The XAMPP Control Panel

Controls for the individual components of your test server can be reached through the XAMPP Control Panel. **The clear user interface** logs all actions and allows you to start or stop individual modules with a single. The XAMPP Control Panel also offers you various other buttons, including:

- **Config:** allows you to configure the XAMPP as well as the individual components
- **Netstat:** shows all running processes on the local computer
- **Shell:** opens a UNIX shell
- **Explorer:** opens the XAMPP folder in Windows Explorer
- **Services:** shows all services currently running in the background
- **Help:** offers links to user forums
- **Quit:** closes the XAMPP Control Panel

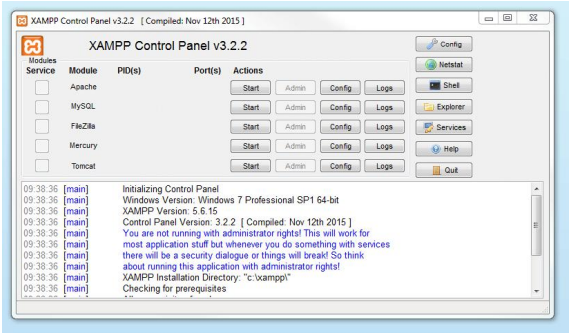

Figure 2.2.8:Main Page of Xampp

In the Control Panel, you can start and stop individual modules Starting modulesIndividual modules can be started or stopped on the XAMPP Control Panel through the corresponding buttons under 'Actions'. You can see which modules have been started because their names are highlighted green under the 'Module' title.

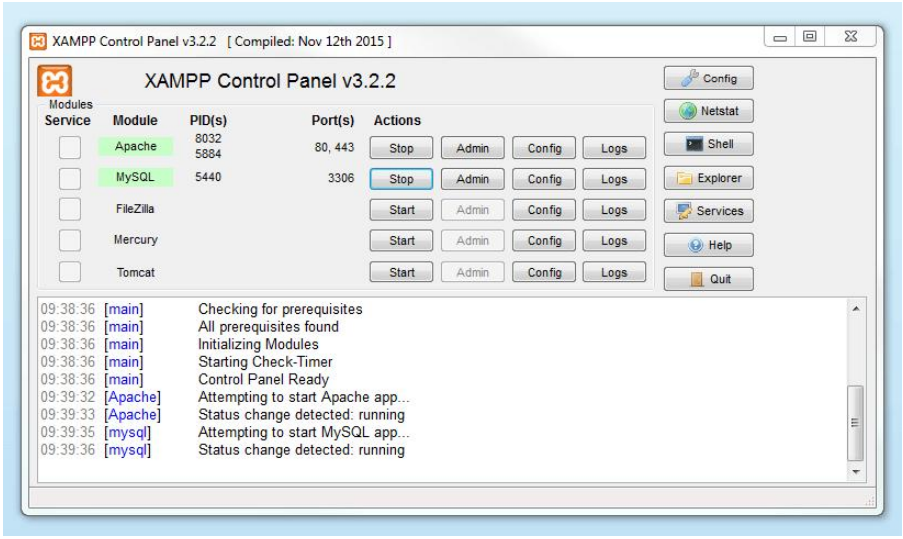

Figure 2.2.9:Selecting the Modules Required .

An active module is marked in green in the Control Panel

If a module can't be started as a result of an error, you'll be informed of this straight away in red font. A **detailed error report** can help you identify the cause of the issue.

Setting up XAMPP

A common source of error connected with Apache is **blocked ports**. If you're using the standard setup, then XAMPP will assign the web server to main port 80 and the SSL port 443. The latter of these particularly is often blocked by other programs. In the example above, it's likely that the Tomcat port is being blocked, meaning the web server can't be started. There are three ways to solve this issue:

- **Change the conflicting port:** Let's assume for the sake of example that the instant messenger program Skype is blocking SSL port 443 (this is a common problem). One way to deal with this issue is to change Skype's port settings. To do this, open the program and navigate via 'Actions', 'Options', and 'Advanced', until you reach the 'Connections' menu. You should find a box checked to allow Skype access to ports 80 and 443. Deselect this checkbox now.

- **Change the XAMPP module port settings**: Click the Config button for the module in question and open the files *httpd.conf* and *httpd-ssl.conf*. Replace port number 80 in *httpd.conf* and port number 443 in *httpd-ssl.conf* with any free ports, before saving the file data. Now click on the general Config button on the right-hand side and select 'Services and Ports Settings'. Customize the ports for the module server to reflect the changes in the *conf* files.

- **End the conflicting program:** The simplest way to avoid port conflicts in the short term is to end the conflicting program (Skype in this case). If you restart Skype after your XAMPP module servers are already running, it will select a different port and your issue will be resolved.

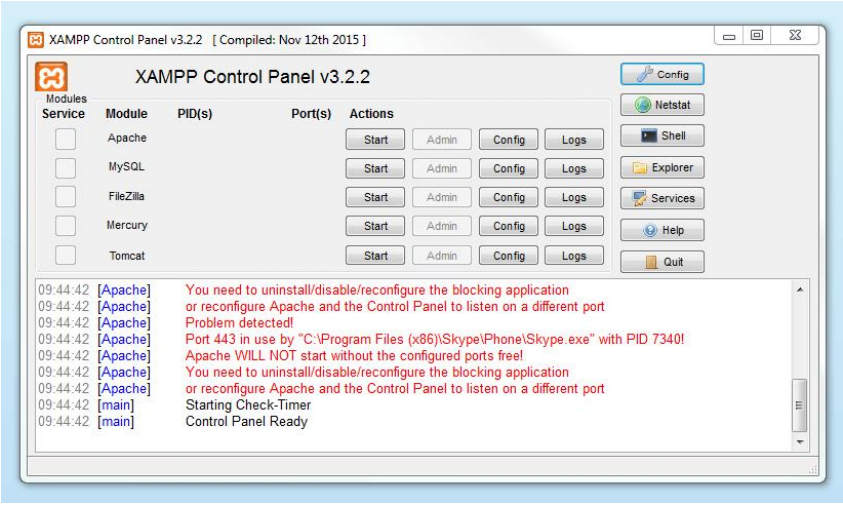

Figure 2.2.10:Conformed Requirements

Modules that can't be started will be shown in red. The user will also receive an error report to

help solve the problem Module administration

You have an 'Admin' option located on the Control Panel for every module in your XAMPP.

- Click on the Admin button of your Apache server to go to the web address of your web server. The Control Panel will now start in your standard browser, and you'll be led to the **dashboard of your XAMPP's local host**. The dashboard features numerous links to websites for useful information as well as the open source project BitNami, which offers you many different applications for your XAMPP, like WordPress or other content management systems. Alternatively, you can reach the dashboard through *localhost/dashboard/*.

Figure 2.2.11: Local Website of Xampp

By clicking on the 'admin' button of the Apache module, the user will be redirected to the local dashboard of XAMPP

- You can use the Admin button of your database module to open **phpMyAdmin**. Here, you can manage the databases of your web projects that you're testing on your XAMPP. Alternatively, you can reach the administration section of your MySQL database via localhost*/phpmyadmin/*.

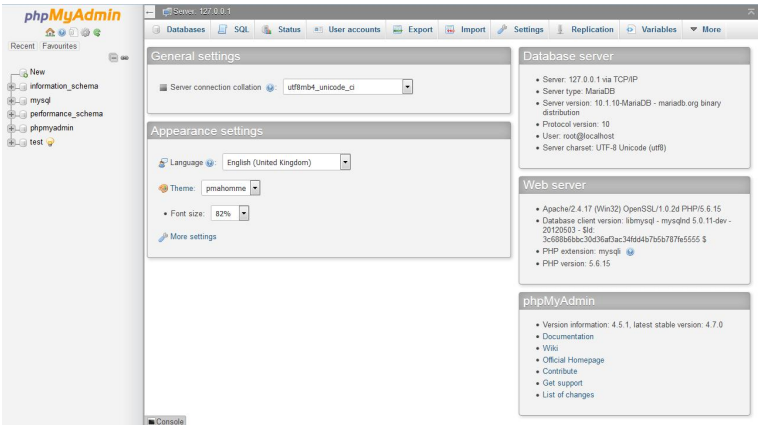

Figure 2.2.12:Database Setup

The web project's databases are managed by the user in phpMyAdmin (accessible via the 'Admin' button in the database module)

Testing your XAMPP installation

To check whether your test server is installed and configured correctly, you have the option to create a **PHP test page**, store them on your XAMPP's local host, and retrieve them via the web browser.

- Open the XAMPP directory through the 'Explorer' button in the Control Panel and choose the folder *htdocs* (C:\xampp\htdocs for standard installations). This directory will store file data collected for web pages that you test on your XAMPP server. The *htdocs* folder should already contain data to help configuration of the web server. But you should store your own projects in a new folder (like 'Test Folder' for example).

- You can create a new PHP page easily by using the following content in your editor and storing it as *test.php* in your ' *test'* folder (C:\xampp\htdocs\test):

```
<html>
 <head>
  <title>PHP-Test</title>
 </head>
 <body>
  <?php echo '<p>Hello World</p>'; ?>
 </body>
</html>
```

- The last step now is to open your web browser and load your PHP page via *localhost/test/test.php*. If your browser window displays the words 'Hello World', then you've successfully installed and configured your XAMPP.

2.3 Browser

Google Chrome is a cross-platform web browser developed by Google. It was first released in 2008 for Microsoft Windows, built with free software components from Apple WebKit and Mozilla Firefox.[15] It was later ported to Linux, macOS, iOS, and Android, where it is the default browser.[16] The browser is also the main component of Chrome OS, where it serves as the platform for web applications.

Most of Chrome's source code comes from Google's free and open-source software project *Chromium*, but Chrome is licensed as proprietary freeware.[14] WebKit was the original rendering engine, but Google eventually forked it to create the Blink engine;[17] all Chrome variants except iOS now use Blink.[18]

As of October 2021, StatCounter estimates that Chrome has a 68% worldwide browser market share (after peaking at 72.38% in November 2018) on personal computers (PC),[19] is

most used on tablets (having surpassed Safari), and is also dominant on smartphones, and at 65% across all platforms combined.[22] Because of this success, Google has expanded the "Chrome" brand name to other products: Chrome OS, Chromecast, Chromebook, Chromebit, Chromebox, and Chromebase.

How To Open Php File In Chrome Using Xampp

simple first step you can Before run/execute any php program server side in the web browser like as a Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, and Apple Safari i have to start a local server service. For that i have to start apache server as well as it can be start by using xampp,wamp,lamp and mamp any types of the server therefor, once our apache service gets started then i go in the browser (Explorer, Firefox, Netscape, and Safari) and enter hostname with project name i.e. **->localhost/project_name/file_name.php** sometimes only localhost is not running properly. therefor for that i have to change some types of the specific configuration setting and after that it runs properly including port number like 8080.

Table:2.1:Softwares and Versions

s.no	Softwares	versions
1	vscode	16.5
2	Xampp	9
3	Browser	Chrome
4	System	windows

Hardware requirements

Chapter 3

Hardware requirements

We strongly recommend a computer fewer than 5 years old.

- Processor: Minimum 1 GHz; Recommended 2GHz or more
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above
- Sound card w/speakers
- Some classes require a camera and microphone

Purpose of the project

Chapter 4

Purpose of the project

4.1 Introduction

Fees payments by students in PSCME College are made through cash deposits, Electronic Funds Transfer (EFT) and Bank drafts to the college's accounts in specific bank branches (PSCMRCET, 2008). For nearly every business, the simple act of collecting payments from consumers is actually quite complex. Organizations want to make it easy and convenient for customers to pay, so they offer multiple choices of payment types and channels". Therefore, the project provides an alternative method that enables secure online fees payment by students and their sponsors.

4.2 Background

PSCMRCET has a large number of students who are supposed to pay all the College fees through cash deposits or bank drafts to the College's accounts in specific bank branches. This method of paying fees has not been efficient enough especially during periods of tests and examinations when most of the students are paying fees to meet the requirements for entering examination rooms. The process of fees payment in such periods is characterized by long queues, too much waiting by students and congestion at banks where payments are made. Students queue to pay fees and those who do not reach counters within the banks‟ working hours are advised to return the next day. This process has always resulted in students missing to sit for their tests and/or examinations while they are queuing to make payments. It has also resulted to too much costs and a lot of time used in transferring and withdrawing money whenever sponsors of students make money transfers to students who can pay College fees at specific bank branches. The process requires sponsors of students from wherever they are to send money to students through either banks, mobile money or any other possible way(s) so that students pay College fees or use EFT that require swift codes to pay fees to the College. This consumes time and sponsors incur extra costs in this process of sending money to students. It was upon this background that the researchers suggested an alternative method which enables secure online fees payment by students and their sponsors.

4.3 Problem Statement

The available modes of fees payment to PSCMRCET through cash deposits, Electronic Funds Transfer (EFT) and bank drafts have caused long queues, students missing to sit for their tests and examinations, and loss of money intended for fees while waiting to reach bank counters to make payments among others. The problem is addressed by developing a system that enables students and their sponsors to securely pay College fees online from wherever they are using credit and debit cards or other UPI payment options.

4.4Main Objective

To develop an online fees payment system that enables students and their sponsors to securely pay College fees online using credit and debit cards.

4.5Specific Objectives

i. To review the existing system used in paying College fees so that its strength and weaknesses are identified.
ii. To design a new system that enables students and their sponsors to pay College fees online from wherever they are using credit and debit cards.
iii. To implement the prototype of the designed system.
iv. To test and validate the system prototype.

4.6 Scope of the Study

The study was carried out in PSCMRCET and was intended to offer an extra channel for fees payment, through the development of a secure online fees payment system. The study focused on the development of a web based system that allows secure online fees payment for PSCMRCET. The system will be used by students and their sponsors to pay all kinds of College fees online, and by College accounts offices to verify students‟ payments. The system captures financial information after payments are made.

System architecture

Chapter 5

System architecture

5.1 System Study

Data that was collected through the use of questionnaires, interviews and in-depth literature review enabled researchers study how fees is paid and what is involved in the process of paying fees to PSCMRCET.

5.1.1 The Existing System

PSCMRCET has an efficient and effective Financial Information System (FINIS) that records, monitors, and reports about student fees payment transactions and other finances in the university. Fees payments by students are made through cash deposits, electronic funds transfer and bank drafts to the university's accounts in specific bank branches and later reflected in FINIS which integrates with the Academic Records Information System (ARIS) that provides information on students. These methods of paying fees have not been efficient enough especially during periods of tests and examinations when most of the students are paying fees to meet the requirements for entering examination rooms. The process of fees payment in such periods is characterized by long queues, too much waiting by students and congestion at banks where payments are made. This process has always resulted in students missing to sit for their tests and/or examinations while they are queuing to make payments. It has also resulted to too much costs and a lot of time used in transferring and withdrawing money whenever sponsors of students make money transfers to students to enable pay university fees at specific bank branches.

5.1.2 The Proposed System

PSCMR-OFPS will provide an additional channel for fees payment to the university online and shall integrate with the already existing systems, FINIS and ARIS. PSCMR-OFPS was proposed to solve the problems associated with the current fees payment methods in the university. MUKOFPS is an alternative platform that enables students and their sponsors to securely pay university fees online using credit and debit cards from wherever they are. The system helps to reduce the number of students that currently miss sitting for their tests and examinations while waiting to reach bank counters to make payments. Sponsors of students, especially those abroad, will also save money and time since it will no longer necessitate them to first transfer money to students before it is paid to the university.

5.2 System Analysis

This presents the analysis of user, functional and non-functional requirements that guided the design and implementation of PSCMR-OFPS.

5.2.1 User Requirements

The major users of the system include students, student sponsors and finance officers in the university finance department. Their requirements include the following;
Students/sponsors shall be able to input transaction information on a user interface that accepts them.
Students/sponsors shall be able to complete fees payment transactions online.
Students/sponsors shall receive feedback that relates the process of online fees payment
Students/sponsors shall be able to view and print or save proof of payment whenever fees payment transactions are successful.
Finance officers shall provide authentication credentials to be able to use the system securely.
Finance officers shall be able to perform searches on details of online payments made for students.
Finance officers shall be able to view summarized reports on all payments made through the system.
Finance officers shall be able to view fees payments in an editable format.

5.2.2 Functional Requirements

i. The system shall accept valid input of registered students‟ payment details from users intending to pay fees online.
ii. The system shall process fees payment transactions so that student fees accounts are credited with the specified amount in each transaction.
iii. The system shall communicate fees payment details for each transaction to the university financial information system, FINIS.
iv. The system shall produce a receipt as a proof of payment for every fees payment transaction made.
v. The system shall provide access to information about how to make payments online.
vi. The system shall produce a listing of transaction information to the finance officers.
vii. The system shall provide feedback to the student describing the status of the transaction.
viii. The system shall be able to generate payment reports to finance officers.

5.2.3 Non-Functional Requirements

i. The system should be easy to maintain.
ii. The system should be compatible with different platforms.
iii. The system should be fast as customers always need speed.
iv. The system should always be available online all times.
v. The system should be secure.
vi. The system should be accessible to online users.
vii. The system should be easy to learn by both sophisticated and novice users.
viii. The system should provide easy, navigable and user friendly interfaces. ix. The system should produce reports in different forms such as tables and graphs for easy visualization by management.
x. The system should have a standard graphical user interface that allows for the on-line data entry, editing, and deleting of data with much ease.

5.3 System Design

The PSCMR-OFPS system design defines the architecture, components/subsystems, modules, interfaces and data required of the system to satisfy specified requirements. In system design the following tools and techniques were used; process modeling, architectural design, data modeling and database design.

5.3.1 Architectural Design

PSCMR-OFPS is a web-based application to be hosted on a web server that communicates to a database server. The user on a web interface makes a web request which is received by the web server. The web server processes the request and interacts with the database server using SQL embedded in PHP scripts. The response is a web page data sent on the web interface for the user.

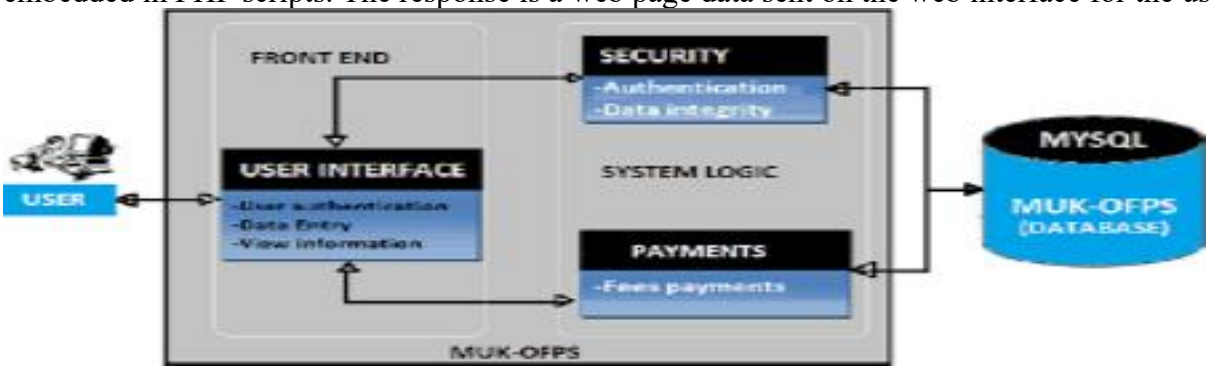

Figure 5.3.1: The Architectural Design for PSCMR-OFPS

5.3.2 Process Modeling

A context diagram and a data flow diagram were used to illustrate the activities that are performed and how data move in PSCMR-OFPS. During process modeling, the following key symbols were used;

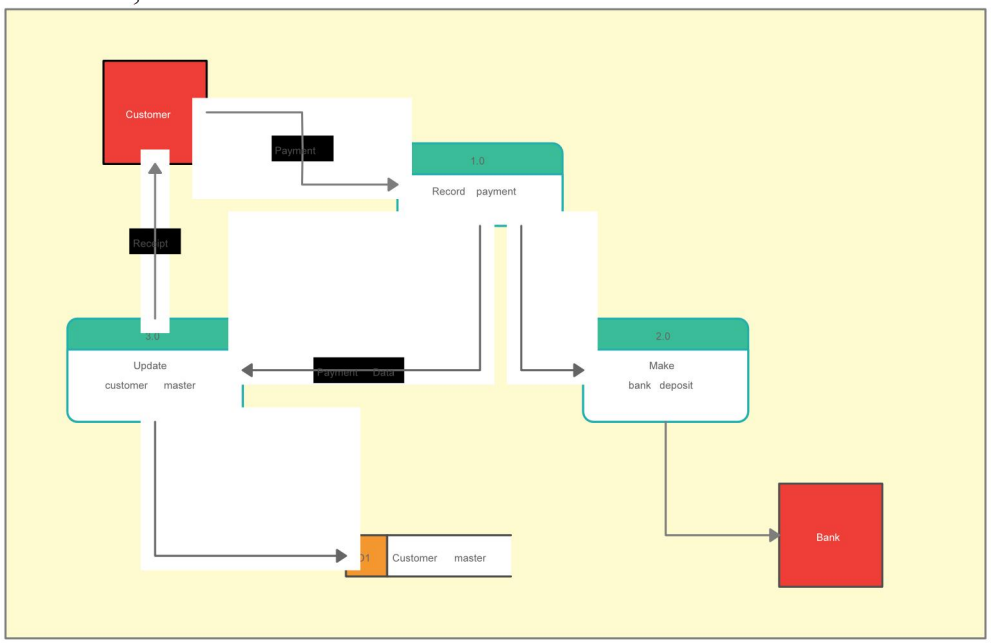

Figure 5.3.2: The context diagram for PSCMR-OFPS

5.3.2.2 Data flow Diagram for PSCMR-OFPS

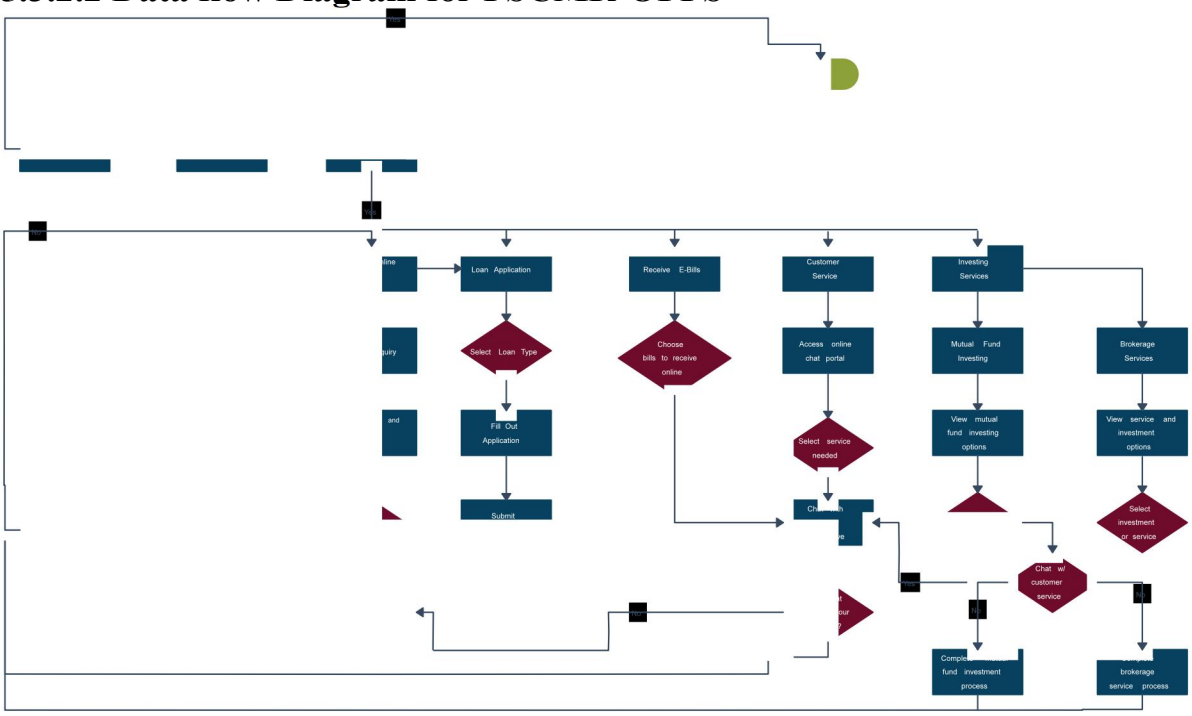

Figure 5.3.3: The Data Flow Diagram for PSCMR-OFPS.

5.4 Data definition of the design objects

The tables below give a description of all design objects that were used in developing the system which include processes, data flows, data stores and the external entities.

Table 5.4. 1: Description of Processes

Process	Description
Validate Payment	Compares user input with details of registered students.
Process Payment	Processes money transfer to the university.
Generate Receipt	Generate receipt as proof a payment transaction.
Authentication	Authenticates users to access the system, and blocks unauthorized access.
Manage Payments Updates	Processes updates to payment records.
Generate Payment Reports	Generates reports as requested by Finance Officers.

Table 5.4.2: Description of Entities

Entity	Description
Student	A student for whom a payment is made. She/he also makes a payment transaction.
Sponsor	Pays fees online for a student.
Finance Officer	Views reports about online payment transactions and updates payment records whenever necessary.
FINIS	Receives payment records for online transactions. Also provides information about finance officers.
ARIS	Academic Records Information System that provides information about registered students.

Table 5.4.3: Description of Data Stores

Data Store	Description
Payment Records	Stores transaction details for online fees payments made for students.

5.5 Data Modeling 4.3.3.1 Identification of Entities and the associated attributes Table 5. 5.1: Entities and their attributes

Payment	PaymentID	Unique identifier of a payment
	Paydate	Date of payment
	Amount	Amount paid
Student	StudentNo	Student Number, unique student identification number
	RegNo	Student registration number
	Fname	Student first name
	Lname	Student last name
PaymentUpdate	TimeStamp	Date and time when an update is made to a payment
FinanceOfficer	OfficerID	Finance officer ID number, a unique identifier of finance officers
	Fname	Finance officer first name
	Lname	Finance officer last name
	Username	Username
	Password	Password

5.5.2 Entity Relationship Diagram (ERD) for PSCMR-OFPS

An ERD was used to show the relationships between the entities involved in the system together with their attributes and indicate the number of occurrences an entity can exist for a single occurrence of the related entity.

Implementation

Chapter 6

Implementation

6.1 Source code for home page:

```
<html>

<head>

    <title>college website</title>

    <link rel="stylesheet" href="boderanimi.css">

    <link rel="stylesheet" href="style.css" />

   <link
href="https://fonts.googleapis.com/css2?family=IBM+Plex+Sans&display=swap"
rel="stylesheet">

<style>

        main {

            width: 100%;

            height: 80vh;

            display: flex;

            justify-content: center;

            align-items: center;

            text-align: center;

        }

    </style>

</head>
```

```
<body>
   <main><section>
         <div class="block">
            </block> <br><br><br><h1>
            <div class="change_text" style="color:yellow;"><b><i><u> TO ONLINE FEE PAYMENT SYSTEM</u></i></b><br><br></div></h1>
               <div style="color:yellow;">
                  <p>YOU MAY PROCEED NOW!!!</p><br>
                  <a href="college.php">
                     <button class="button">PROCEED</button></a>
              </div></section>
   </main>
</body>
</html>
```

Declaration:

This is the main home page that everyone can make their online fee through this website.

Output for the home page:

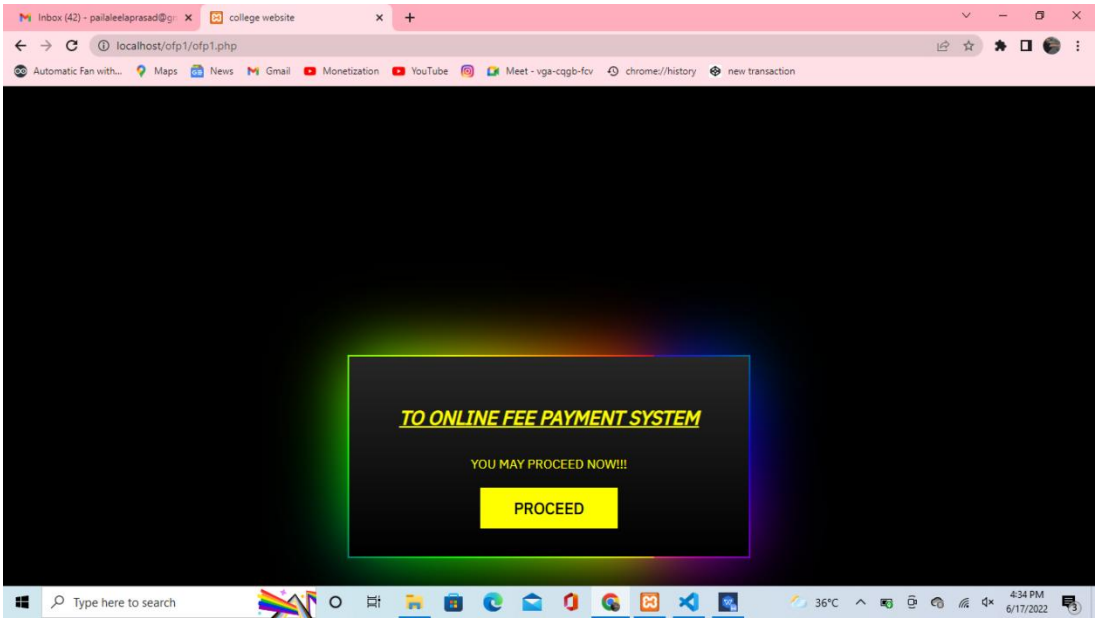

Figure:6.1:Home Page

6.2 Source code for college selection page:

```
<html>
<head>
    <title>college</title>
    <link rel="stylesheet" href="login.css" />
</head>
<body>
    <div class="login-box">
        <h2 style="color:yellow"><u><i>Select College</i></u></h2>
        <form>
            <div class="user-box">
```

```
            <label>State</label><br><br>

            <select name="state" id="state" class="form-control"
style="background-color:blue;">

               <option value="Andhra Pradesh">Andhra Pradesh</option>

            </select><br>

         </div>

         <div class="user-box">

            <label>Institution</label><br><br>

            <form>

               <button style="color:red">*...SELECT ANY BUTTON...*</button>

            </form><br><br>

            <form action="login.php">

               <button style="background-color:white; border-color:blue; color:red"
value="PSCMRCET">PSCMRCET</button>

            </form>

            <form action="https://feepay.narayanagroup.com/">

               <button style="background-color:white; border-color:blue; color:red"
value="NARYANA">NARAYANA</button>

            </form>

            <form action="https://aec.edu.in/?p=Refund">

               <button style="background-color:white; border-color:blue; color:red"
value="ADITHYA">ADITHYA</button>

            </form>
```

```
<form action="others_college.PHP">

  <button value="OTHER" style="background-color:white; border-color:blue; color:red">OTHER</button>

</form>

<form action="https://paytm.com/education">

  <button value="OTHER" style="background-color:white; border-color:blue; color:red">Pay Through Paytm</button>

</form>

</script></div>

</form></div>

</body>

</html>
```

Declaration:

This is the page that you can select the college that which specific college you have to pay through it.

Result:

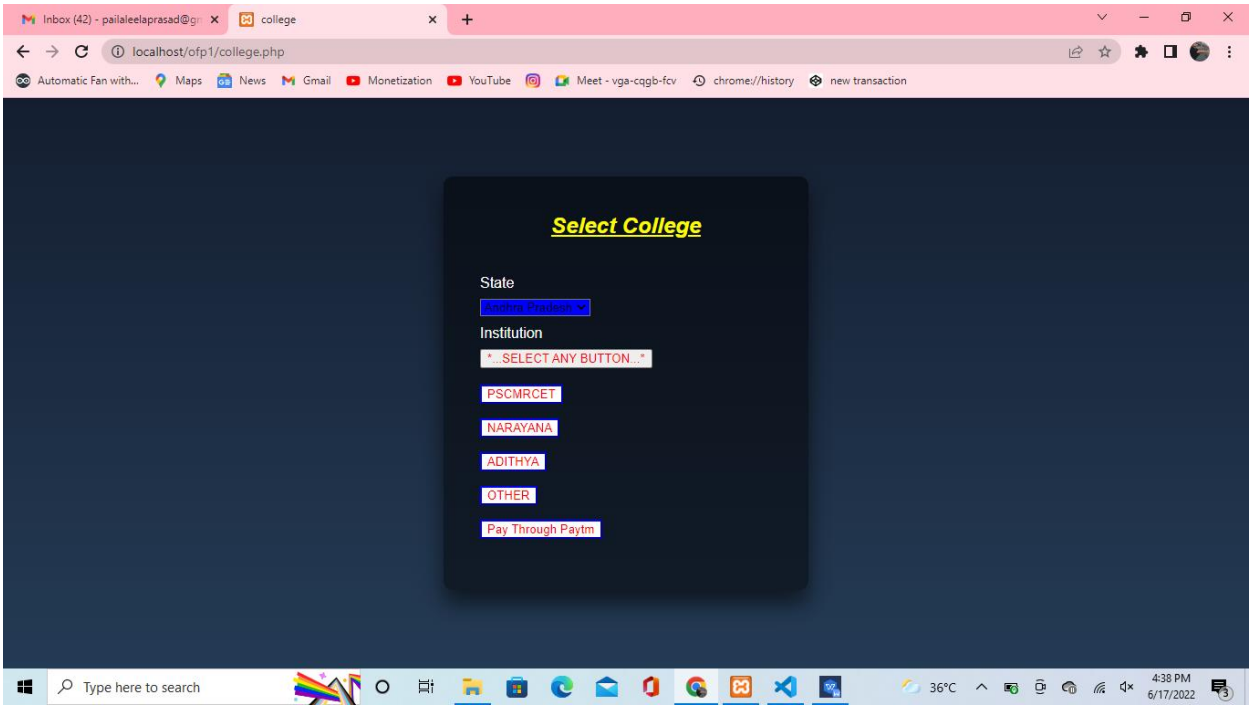

Figure 6.2: Selecting college

6.3 Source code for login:

```
<body>
   <div class="login-box">
      <h2>Login</h2>
      <form>
         <div class="user-box">
            <input type="text">
            <label>Username</label>
         </div>
         <div class="user-box">
```

```
            <input type="password">
            <label>Password</label>
        </div>
        <a href="pscmr.php">
            <span></span>
            <span></span>
            <span></span>
            <span></span> Submit </a>
    </form> </a><a href="update credentials.php">Update Credentials</a><br><br><a href="regform.php">REGISTER</a>
  </div>
</body>
```

Declaration:

This is the page that everyone can login to my website and their can make their fee payment through my website for any colleges. And also they can update their credentials that which are integrated with the database of my websites. If they have no account in my site they have to register first with the help of a given button for that so the users can register with their personal credentials with the help of it.

Result:

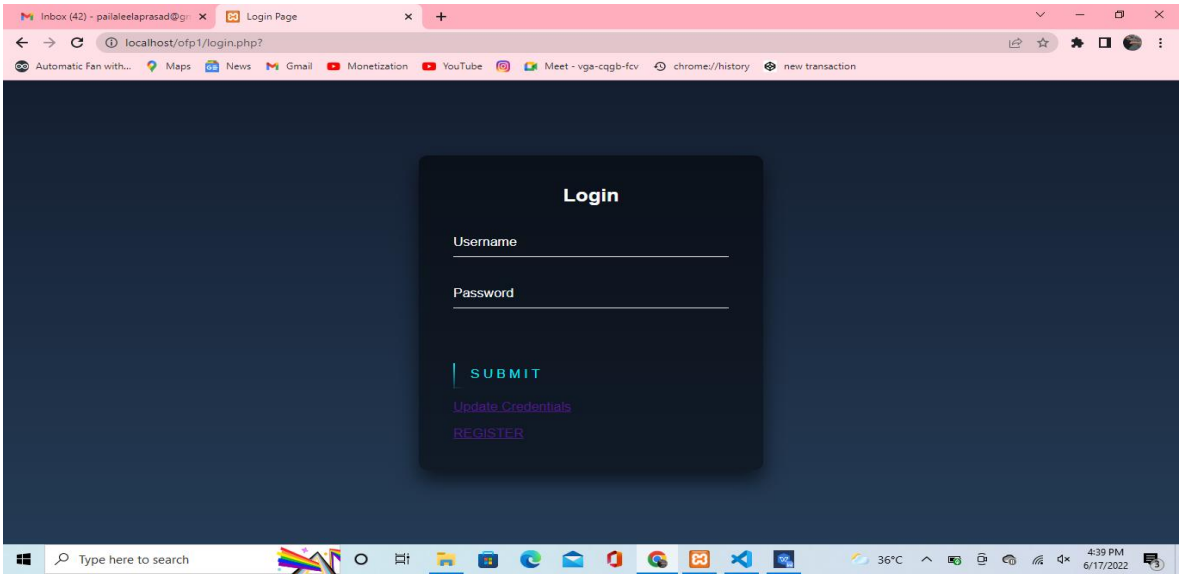

Figure 6.3: Login Page

6.4 Source code for college web site:

```
<body>

   <div>

      <img src="banner.jpg" align="left" height="15%" width="100%">

    </div> <hr> <br> <div style="border-radius: 10px; background: rgb(213, 212, 212); padding: 40px;" align="center">

 <div class="row"><div class="col-md-6 mb-4">

<form class="form-inline md-form mr-auto" method="POST">

                Student Number:    

                 <input class="form-control mr-sm-2" placeholder="Student Number" aria-label="Search" type="text" name="search">

                  <button class="fa fa-search" type="submit" style="color:rgb(255, 0, 0)" align="center">Search</button>

            </form></div></div></div></body>
```

Declaration:

This is the page that you can enter through the register number that you can enter through your user name

Result:

Figure 6.4:Student Login Of PSCMR

6.5 Source code for terms and conditions page;

```
<body bgcolor="black" text="yellow">

  <div class="wrapper"><h2>WELCOME ,

  <?php echo $_SESSION['sess_user']; ?></h2>

  </div> <h1>Online Fee Policies</h1> <p> <i><u><b><h3>Privacy Policy:</h3></b></u></i><br> Terna College of Engineering considers the protection of your personal information important and shall take necessary care to safeguard your privacy. If you decide to access the website, your visit and any dispute over privacy are subject to this Privacy Policy and Our Terms and Conditions of use. Our Policy regarding the collection, use, and disclosure, if any, of personal information is very strict and we adhere to the best of practices to guard your personal information with care. </p><p><u><b><h3>Terms and Conditions for Online Fee Submission:-</h3></b></u><br><div class="tacbox"><input id="checkbox" type="checkbox" required/> <label
```

for="checkbox"> I agree to these Terms and Conditions.</label>

<a href="Types of fee.php" align="center">

<button id="Enter" >Enter</button>

</body>

Declaration:

This the page that can make you to welcome with the help of your username and gives the full details about the terms and conditions to pay any payment through our website. So you can get awareness about the payment methods and what are the safitys you have to take so make sure that everyone has to read the terms and conditions before you are paying anything through online process.

Result:

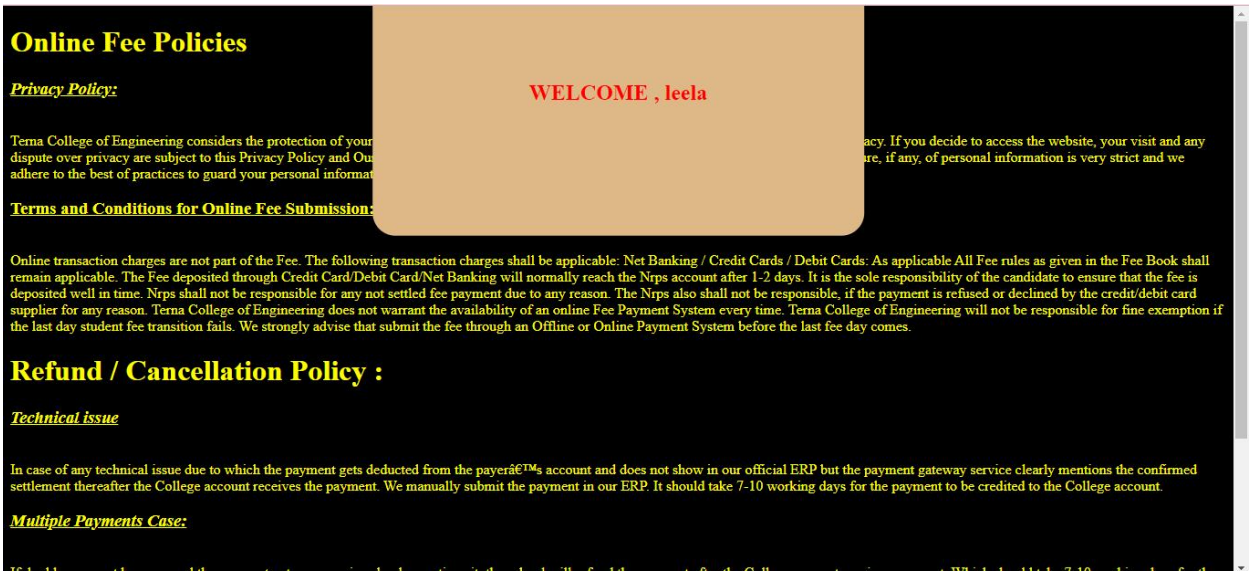

Figure 6.5 Terms and Conditions

6.6 Source code for select fee type:

```
<!DOCTYPE html>

<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <meta http-equiv="X-UA-Compatible" content="ie=edge">

   <link
href="https://fonts.googleapis.com/css?family=Lato|Nanum+Gothic:700|Raleway&display=swa
p" rel="stylesheet">
   <link rel="stylesheet" href="StyleNeon.css">

   <title>Fee payment</title>
</head>

<body>
   <h1>Select Fee Payment</h1>

   <section class="main-container">

      <div class="btn-row">
         <div class="ct-btn">
            <a href="college_fee.php">
               <button class="sky-neon">College Fee</button></a>
         </div>
         <div class="ct-btn">
            <a href="JVD.php">
               <button class="breath-neon">JVD Return</button>
            </a>
         </div>
         <div class="ct-btn">
            <a href="Campus fee.php">
               <button class="heart-bit-neon">Campus Fee</button>
            </a>
         </div>
      </div>

      <div class="btn-row">
         <div class="ct-btn">
            <a href="Exam Fees.php">
               <button class="border-neon">Exam Fee</button>
            </a>
```

```
        </div>
        <div class="ct-btn">
          <a href="Supply Fee.php">
            <button class="inner-height-neon"><span class="bg-height"></span><span
class="btn-text">Supply
                Fee</span></button>
          </a>
        </div>
        <div class="ct-btn">
          <a href="CRT.php">
            <button class="inner-width-neon"><span class="bg-width"></span><span
class="btn-text">CRT
                Fee</span></button></a>
        </div>
      </div>

      <div class="btn-row">
        <div class="ct-btn">
          <div class="gr-solid">
            <a href="Bus Fee.php">
              <span class="gr-shadow1"></span>
              <button class="gradient-neon">Bus Fee</button>
            </a>
          </div>
        </div>
        <div class="ct-btn">
          <div class="gr-move">
            <a href="Hostel Fee.php">
              <span class="gr-shadow2"></span>
              <button class="gradient-neon">Hostel Fee</button>
            </a>
          </div>
        </div>
        <div class="ct-btn">
          <div class="gr-tri">
            <a href="Other's.php">
              <span class="gr-shadow3"></span>
              <button class="gradient-neon">Other's</button>
            </a>
          </div>
        </div>
      </div>
    </section>
</body>
```

Declaration:

This is the page that you have to select which type of payment you are doing with this website.That you have to select from here.

Result:

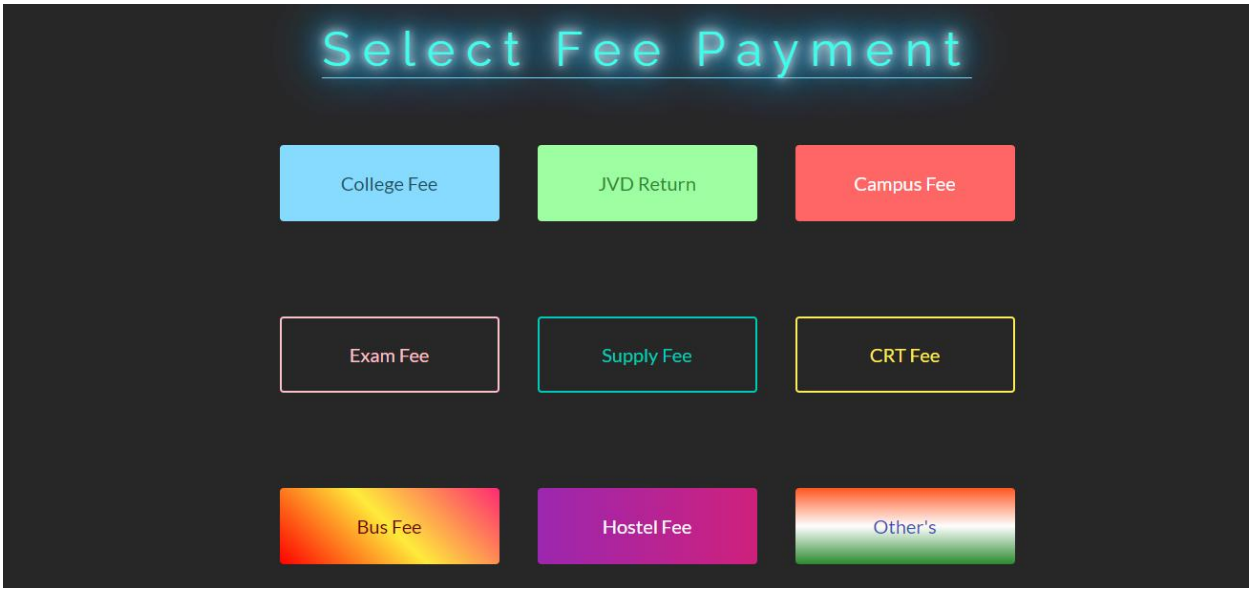

Figure 6.6: Selecting the type of payment

6.7 Source code for fee payment registration page:

```
<html>
<head>
   <link rel="stylesheet" href="pay.css">
   <title>Campus fee</title>
</head>
<body style="color:white;">
   <h1 align="center"><b><u>Payment Detail's</u></b></h1>
   <h3>
      Fee:
      <input type="text" name="Campus Fees" value="Campus Fees" readonly><br>
      <br>
      <div class='row_accord'><label>
            REG NO :
         </label>
         <input type='Text' name='outref11' placeholder="enter redg no." maxLength='75' value='' id='outref11' />
```

```
</div>
<input type='hidden' name='controls' value='outref11' id='controls' />
<input type='hidden' name='validateMandatory' id='validateMandatory' value='outref11#SEP#REG NO#SEP#(^[0-9a-zA-Z&_. /@-]{1,76}$)#SEP#REG NO Should be valid.' />
<!--Populating fields from control tag-->
<div class='row_accord'><label>
    <br>STUDENT NAME :
  </label>
  <input type='Text' name='outref12' maxLength='75' value='' id='outref12' placeholder="Enter Your Name" />
</div>
<input type='hidden' name='controls' value='outref12' id='controls' />
<input type='hidden' name='validateMandatory' id='validateMandatory' value='outref12#SEP#STUDENT NAME#SEP#(^[0-9a-zA-Z&_. /@-]{1,76}$)#SEP#STUDENT NAME Should be valid.' />
<!--Populating fields from control tag-->
<div class='row_accord'><label>
    <br>FATHER NAME :
  </label>
  <input type='Text' name='outref13' maxLength='75' value='' id='outref13' placeholder="Enter Your Father Name" />
</div>
<input type='hidden' name='controls' value='outref13' id='controls' />
<input type='hidden' name='validateNotMandatory' id='validateNotMandatory' value='outref13#SEP#FATHER NAME#SEP#(^[0-9a-zA-Z&_. /@-]{1,76}$)#SEP#FATHER NAME Should be valid.' />
<!--Populating fields from control tag-->
<div class='row_accord'><label>
    <br>COURSE :
  </label>
  <span class="plain-select-small">
    <select name='outref14' id='outref14'>
      <option value=''>--Select COURSE--</option>
      <option value='B.tech'>B.Tech</option>
      <option value='other'>Other</option>
    </select>
  </span>
</div>
```

```
<input type='hidden' name='controls' value='outref14' id='controls' />
<input type='hidden' name='validateMandatory' id='validateMandatory' value='outref14#SEP#COURSE#SEP#null#SEP#null' />
<!--Populating fields from control tag-->
<div class='row_accord'><label>
    <br>SEMISTER :
  </label>
  <span class="plain-select-small">
    <select name='outref15' id='outref15'>
      <option value="">--Select SEMISTER--</option>
      <option value='I'>I</option>
      <option value='II'>II</option>
      <option value='III'>III</option>
      <option value='IV'>IV</option>
      <option value='V'>IV</option>
      <option value='VI'>IV</option>
      <option value='VII'>IV</option>
      <option value='VIII'>IV</option>
    </select>
  </span>
</div>
<input type='hidden' name='controls' value='outref15' id='controls' />
<input type='hidden' name='validateMandatory' id='validateMandatory' value='outref15#SEP#SEMISTER#SEP#null#SEP#null' />
<!--Populating fields from control tag-->
<div class='row_accord'><label>
    <br>AMOUNT :
  </label>
  <input type='Variable' name='outref16' maxLength='10' value='' id='outref16' / placeholder="Enter Amount"><br><br>
</div>
<input type='hidden' name='controls' value='outref16' id='controls' />
<input type='hidden' name='validateMandatory' id='validateMandatory' value='outref16#SEP#AMOUNT#SEP#(^(\d{1,20})(\.[0-9][0-9])?$)#SEP#AMOUNT Should be valid amount' /> Remarks:
<textarea id="tutorial" name="tutorial" rows="5" cols="40"></textarea>
<div class="panel panel-default">
  <div class="panel-heading">
    <h4 class="panel-title">
      Please enter your Name, Date of Birth <b>(For Personal
```

```
                Banking) / Incorporation (For Corporate
Banking)</b>  & Mobile Number.
                <br>This is required to reprint your e-receipt / remittance(PAP) form,
if the need arises.
              </h4>
            </div>
            <div class="panel-collapse">
              <div class="panel-body">
                <div class="row_accord">
                  <label>Name :</label> <input type="text" name="cusName"
id="cusName" maxlength="30" value="" placeholder="Enter Your Name"
/> 
                  <span id="nameerrmsg"></span>
                </div>
                <br>
                <div class="row_accord">
                  <!-- <label>Date of Birth / Incorporation:</label> -->
                  <label>Date Of Birth / Incorporation :</label> 
                  <!--  EMRO ends -->
                  <input type="date" name="dateOfBirth" id="dateOfBirth"
maxlength="12" placeholder="Enter date of birth" />
                  <span id="doberrmsg"></span>
                </div>
                <br>
                <div class="row_accord">
                  <label>Mobile Number :</label> 
                  <input type="text" name="mobileNo" id="mobileNo"
maxlength="10" class="numbersOnly" value="" placeholder="Enter Mobile
Number" />
                  <span id="moberrmsg"></span>
                </div><br>
                <!--   Added for Payer email alert Start -->
                <div class="row_accord">
                  <label>Email Id: </label> 
                  <input type="text" name="emailId" id="emailId" maxlength="200"
value="" placeholder="Enter Your Mail Id" />
                  <span id="emailerrmsg"></span>
                </div><br>
                <!--   Added for Payer email alert End -->
```

```
            </div>
        </div>
        <div class="home_btn" align="center">
            <div class="ct-btn">
                <a href="payment page.php">
                    <button class="Submit"><span class="bg-height"></span><span
class="btn-text">Pay</span></button>
                </a>
            </div>
            <div class="ct-btn">
                <a href="Campus fee.php">
                    <button class="Reset"><span class="bg-height"></span><span
class="btn-text">Reset</span></button>
                </a>
            </div>
            <div class="ct-btn">
                <a href="Types of fee.php">
                    <button class="Cancel"><span class="bg-height"></span><span
class="btn-text">Back</span></button>
                </a>
            </div>
        </div>
        <form>
            <form>
                <div class="notifications" style="color:white;">
                    <ul>
                        <li>Mandatory fields are marked with an asterisk (*)</li>
                        <li>The payment structure document if available will contain
detailed instructions about the online payment process.</li>
                        <li>Date specified(if any) should be in the format of 'ddmmyyyy'.
Eg., 02082008</li>
                        <li>For Amount fields, only numbers are allowed and for free
text fields (mandatory), following special characters are allowed: . / @ - _ &</li>
                    </ul>
                </div>
            </form>
        </form>
    </h3>
```

</body>
</html>

Declaration:

This the page the you have to fill the details for the payment of your page has to require the details of yours and fill the details that are initiated their and enter the pay button. If you fill it and you entered then it has no way to come your money back. So be careful when you are filling the forms.

Result:

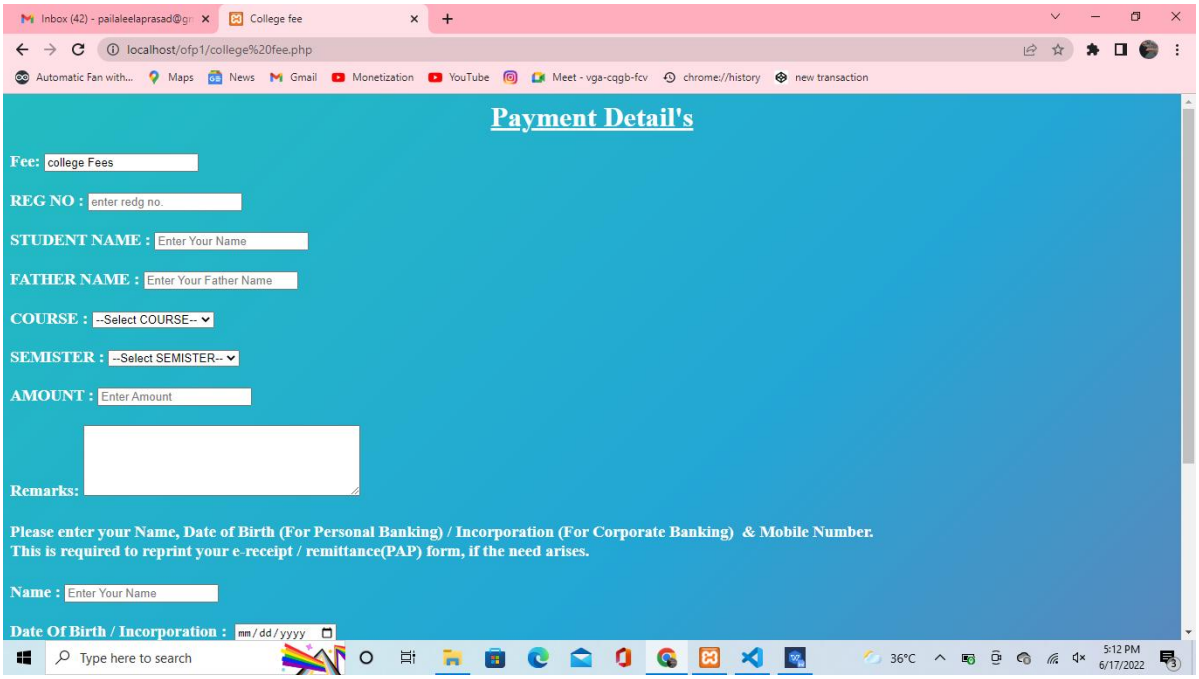

Figure 6.7:Payment Details Page

6.8 Source code for payment page:

```
<!doctype html>

<html>

<head>

   <meta charset='utf-8'>

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

```
    <title>Payment page</title>

    <link
href='https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css'
rel='stylesheet'>

    <link                href='https://use.fontawesome.com/releases/v5.8.1/css/all.css'
rel='stylesheet'>

    <script                                                          type='text/javascript'
src='https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js'></script>

    <style>

        ::-webkit-scrollbar {

            width: 8px;

        }

        /* Track */

        ::-webkit-scrollbar-track {

            background: #f1f1f1;

        }

        /* Handle */

        ::-webkit-scrollbar-thumb {

            background: #888;

        }

        /* Handle on hover */


        ::-webkit-scrollbar-thumb:hover {
```

```
        background: #555;
    }
    body {
        background: #f5f5f5
    }
    .rounded {
        border-radius: 1rem
    }
    .nav-pills .nav-link {
        color: #555
    }
    .nav-pills .nav-link.active {
        color: white
    }
    input[type="radio"] {
        margin-right: 5px
    }
    .bold {
        font-weight: bold
    }
</style>
```

```
</head>
<body className='snippet-body'>
   <div class="container py-5">
      <!-- For demo purpose -->
      <div class="row mb-4">
         <div class="col-lg-8 mx-auto text-center">
            <h1 class="display-6">Online Fee Payment Forms</h1>
         </div>
      </div>
      <!-- End -->
      <div class="row">
         <div class="col-lg-6 mx-auto">
            <div class="card ">
               <div class="card-header">
                  <div class="bg-white shadow-sm pt-4 pl-2 pr-2 pb-2">
                     <!-- Credit card form tabs -->
                     <ul role="tablist" class="nav bg-light nav-pills rounded nav-fill mb-3">
                        <li class="nav-item">
                           <a data-toggle="pill" href="#credit-card" class="nav-link active "> <i class="fas fa-credit-card mr-2"></i> Credit Card
                           </a>
```

```
                    </li>

                    <li class="nav-item">

                        <a data-toggle="pill" href="#UPI" class="nav-link "> <i class="fab fa-UPI mr-2"></i> UPI </a>

                    </li>

                    <li class="nav-item">

                        <a data-toggle="pill" href="#net-banking" class="nav-link "> <i class="fas fa-mobile-alt mr-2"></i> Net Banking </a>

                    </li>

                </ul>

            </div>

            <!-- End -->

            <!-- Credit card form content -->

            <div class="tab-content">

                <!-- credit card info-->

                <div id="credit-card" class="tab-pane fade show active pt-3">

                    <form role="form" onsubmit="event.preventDefault()">

                        <div class="form-group"> <label for="username">

                            <h6>Card Owner</h6>

                        </label> <input type="text" name="username" placeholder="Card Owner Name" required class="form-control "> </div>

                        <div class="form-group"> <label for="cardNumber">

                            <h6>Card number</h6>
```

```
                    </label>

                    <div class="input-group"> <input type="text" name="cardNumber" placeholder="Valid card number" class="form-control " required>

                        <div class="input-group-append"> <span class="input-group-text text-muted">

                            <i class="fab fa-cc-visa mx-1"></i> <i

                                class="fab fa-cc-mastercard mx-1"></i> <i

                                class="fab fa-cc-amex mx-1"></i> </span> </div>

                    </div>

                </div>

                <div class="row">

                    <div class="col-sm-8">

                        <div class="form-group"> <label><span class="hidden-xs">

                            <h6>Expiration Date</h6>

                        </span></label>

                        <div class="input-group"> <input type="number" placeholder="MM" name="" class="form-control" required> <input type="number" placeholder="YY" name="" class="form-control" required> </div>

                    </div>

                </div>

                <div class="col-sm-4">
```

```
                        <div class="form-group mb-4"> <label data-
toggle="tooltip" title="Three digit CV code on the back of your card">

                            <h6>CVV <i class="fa fa-question-circle d-
inline"></i></h6>

                        </label> <input type="text" required class="form-
control"> </div>

                    </div>

                </div>

                <div class="card-footer"> <button type="button"
class="subscribe btn btn-primary btn-block shadow-sm"><a
href="payment_report.php" style="color:white;" > Confirm Payment</a>

                    </button>

                </form>

                </div>

            </div>

            <!-- End -->

            <!-- Paypal info -->

            <div id="UPI" class="tab-pane fade pt-3">

                <h6 class="pb-2">Select your UPI account type</h6>

                <form onchange="myFunction()" name="frmRadio"
id="radio-buttons">

                    <div class="form-group ">

                        <label class="radio-inline"> <input type="radio"
id="googlepay" name="optradio"

                            name="option"
```

```
                    onclick="document.getElementById('radio-
buttons').action='; "> Google

                    Pay</label>

                    <label class="radio-inline"> <input type="radio"
id="paytm" name="optradio"

                    onclick="document.getElementById('radio-
buttons').action='; "

                    class="ml-5">Paytm</label>

                    <label class="radio-inline"> <input type="radio"
name="optradio" id="phonepe"

                    class="ml-5"

                    onclick="document.getElementById('radio-
buttons').action='; ">Phonepe</label>

                    </div>

                    <div id="button"></div>

                    <p class="text-muted"> Note: After clicking on the button,
you will be directed to a secure gateway for payment. After completing the
payment process, you will be redirected back to the website to view details of your
order. </p>

                </div>

                </form>

                <!-- End -->

                <!-- bank transfer info -->

                <div id="net-banking" class="tab-pane fade pt-3">

                  <div class="form-group "> <label for="Select Your Bank">
```

```
                    <h6>Select your Bank</h6>
                </label> <select class="form-control" id="ccmonth">
                    <option value="" selected disabled>--Please select your Bank--</option>
                    <option>SBI</option>
                    <option>Andhra Bank</option>
                    <option>HDFC</option>
                    <option>SBH</option>
                    <option>Karur vysya Bank</option>
                    <option>Karnataka Bank</option>
                    <option>TMB</option>
                    <option>IDFC Bank</option>
                    <option>AXIS Bank</option>
                    <option>Central Bank of India</option>
                </select> </div>
                <div class="form-group">
                    <p> <button type="button" class="btn btn-primary "><i
                            class="fas fa-mobile-alt mr-2"></i> Proceed Payment</button> </p>
                </div>
                <p class="text-muted">Note: After clicking on the button, you will be directed to a secure gateway for payment. After completing the payment process, you will be redirected back to the website to view details of your order. </p>
```

```
                    </div>
                </div>
            </div>
        </div>
    </div>
    <script type='text/javascript' src='https://stackpath.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.bundle.min.js'></script>
    <script type='text/javascript' src='#'></script>
    <script type='text/javascript' src='#'></script>
    <script type='text/javascript'>
        $(function() {
            $('[data-toggle="tooltip"]').tooltip()
        })
    </script>
    <script>
        function myFunction() {
            if (document.getElementById("googlepay").checked) {
                document.getElementById("button").innerHTML = "<p> <a href='https://pay.google.com/gp/w/u/0/home/signup?sctid=8298021898220379'><button type='button' class='btn btn-primary'><i class='fab fa-UPI mr-2'></i>Proceed </button></a> </p>";
            }
            if (document.getElementById("paytm").checked) {
```

```
                document.getElementById("button").innerHTML = "<p> <a href='https://paytm.com/'><button type='button' class='btn btn-primary'><i class='fab fa-UPI mr-2'></i>Proceed </button></a> </p>";

            }

            if (document.getElementById("phonepe").checked) {

                document.getElementById("button").innerHTML = "<p> <a href='https://phonepe.com/'><button type='button' class='btn btn-primary'><i class='fab fa-UPI mr-2'></i>Proceed </button></a> </p>";

            }

        }

    </script>

</body>

</html>
```

Declaration:

This is the page of your payment forms that you have to select the which type of the payment that you are want to pay you have gone through it and make your payment successful.

Result:

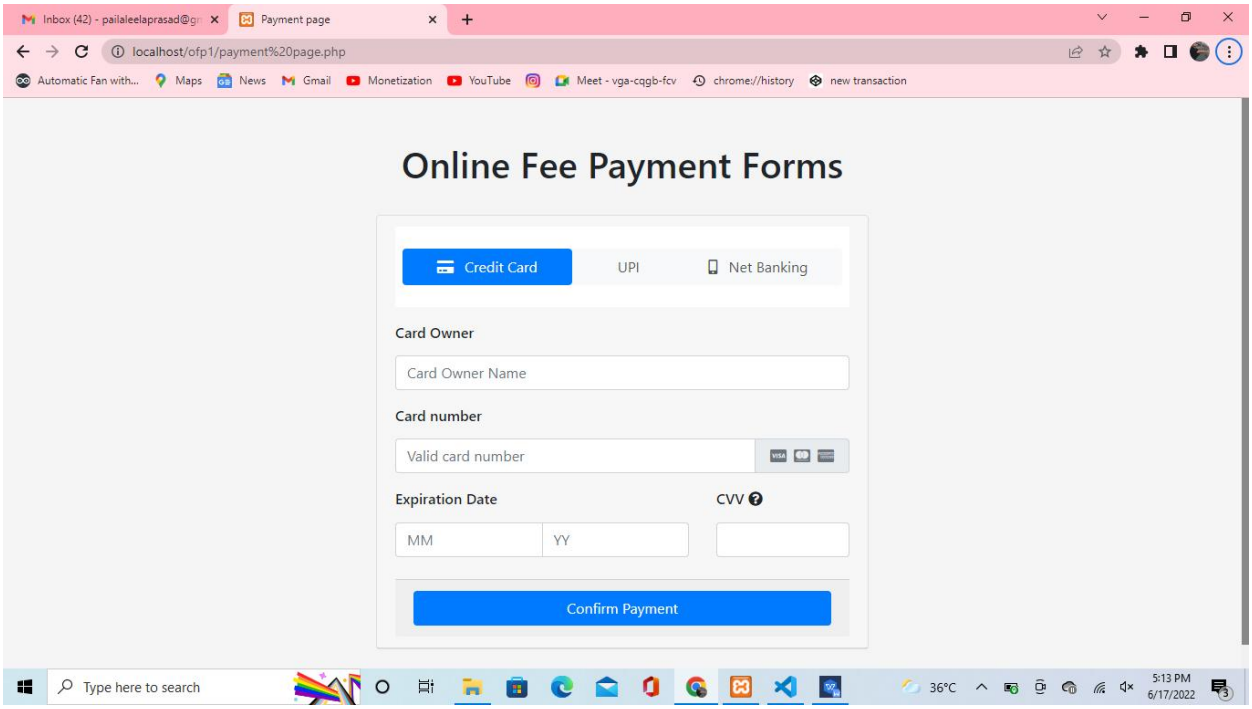

Figure 6.8:Online Fee Payment Forms

6.9 Source code for payment receipt:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

<html style="-moz-osx-font-smoothing: grayscale; -webkit-font-smoothing: antialiased; background-color: #464646; margin: 0; padding: 0;">

<head>

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

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<meta name="format-detection" content="telephone=no">

<title>Payment successful RECEIPT</title>

```
<script>
    function printDocument(elemid) {
        if (elemid == "") {
            window.print();
        } else {
            var arrelemid = elemid.split(',');
            var htmlContent = "";
            for (var i = 0; i < arrelemid.length; i++) {
                htmlContent += document.getElementById(arrelemid[i]).innerHTML;
            }
            var ww = screen.availWidth;
            var wh = screen.availHeight - 90;
            var pw = window.open("", "newWin", "width=" + ww + ",height=" +
wh);
            pw.document.write('<html><title>Printed Page</title><body>');
            pw.document.write('</head><body>');
            pw.document.write(htmlContent);
            pw.document.write('</body></html>');
            pw.document.close();
            pw.print();
            pw.close();
        }
```

```
    }
  </script>
</head>
<body bgcolor="#d7d7d7" class="generic-template" style="-moz-osx-font-smoothing: grayscale; -webkit-font-smoothing: antialiased; background-color: #d7d7d7; margin: 0; padding: 0;">
  <!-- Content Start -->
  <table cellpadding="0" cellspacing="0" cols="1" bgcolor="#d7d7d7" align="center" style="max-width: 600px;">
    <tr bgcolor="#d7d7d7">
      <td height="50" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">
      </td>
    </tr>
    <!-- This encapsulation is required to ensure correct rendering on Windows 10 Mail app. -->
    <tr bgcolor="#d7d7d7">
      <td style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">
        <!-- Seperator Start -->
        <table cellpadding="0" cellspacing="0" cols="1" bgcolor="#d7d7d7" align="center" style="max-width: 600px; width: 100%;">
          <tr bgcolor="#d7d7d7">
            <td height="30" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align:
```

```
top;">
                    </td>
                </tr>
            </table>
            <!-- Seperator End -->
            <!-- Generic Pod Left Aligned with Price breakdown Start -->
            <table align="center" cellpadding="0" cellspacing="0" cols="3" bgcolor="white" class="bordered-left-right" style="border-left: 10px solid #d7d7d7; border-right: 10px solid #d7d7d7; max-width: 600px; width: 100%;">
                <tr height="50">
                    <td colspan="3" style="color: #09ff00; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">
                    </td>
                </tr>
                <tr align="center">
                    <td width="36" style="color: #00ff00; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">
                    </td>
                    <td class="text-primary" style="color: #22f12c; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">
                        <img
src="http://dgtlmrktng.s3.amazonaws.com/go/emails/generic-email-template/tick.png" alt="GO" width="50" style="border: 0; font-size: 0; margin: 0; max-width: 100%; padding: 0;">
```

```
                </td>

                <td width="36" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">

                </td>

            </tr>

            <tr height="17">

                <td colspan="3" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">

                </td>

            </tr>

            <tr align="center">

                <td width="36" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">

                </td>

                <td class="text-primary" style="color: #29f122; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">

                    <h1 style="color: #1070ff; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 30px; font-weight: 700; line-height: 34px; margin-bottom: 0; margin-top: 0;">

                        Payment received</h1>

                </td>

                <td width="36" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align:
```

```
top;">
                </td>
            </tr>
            <tr height="30">
                <td colspan="3" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">
                </td>
            </tr>
            <tr align="left">
                <td width="36" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">
                </td>
                <td style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">
                    <p style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 22px; margin: 0;">
                        Hi [name],
                    </p>
                </td>
                <td width="36" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">
                </td>
```

```
</tr>

<tr height="10">

<td colspan="3" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">

</td>

</tr>

<tr align="left">

<td width="36" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">

</td>

<td style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">

<p style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 22px; margin: 0;">

Your transaction was successful!</p>

<br>

<p style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 22px; margin: 0; ">

<strong>Payment Details:</strong><br />

Bank: <br />

Amount: <br />

Account: <br />

Mobile Number: <br />
```

```
                Email Id: <br />

                Transaction ID: <br />

            </p>

            <br>

            <hr><br>

            <p style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 22px; margin: 0;">

                We advise to keep this email & Transaction ID for future reference.    <br /></p>

            </td>

            <td width="36" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">

            </td>

        </tr>

        <tr height="30">

            <td style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">

            </td>

            <td style="border-bottom: 1px solid #D3D1D1; color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">

            </td>

            <td style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">
```

```
                </td>

            </tr>

            <tr height="30">

                <td colspan="3" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">

                </td>

            </tr>

            <tr align="center">

                <td width="36" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">

                </td>

                <td style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">

                    <p style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 22px; margin: 0;">

                        <strong>Transaction reference: ${authorizationCode}</strong>

                    </p>

                    <p style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 22px; margin: 0;">

                        Transaction date: [time] [date]</p>

                    <input type='button' id='btn' value='PRINT' onclick='printDocument("");'><BR><BR>

                    <U><a href="ofp1.php"><input type='button' id='btn' value='GO
```

```
Back To Home'></a></U>

                    <p style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 22px; margin: 0;">

                    </p>

                </td>

                <td width="36" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">

                </td>

            </tr>



            <tr height="50">

                <td colspan="3" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align: top;">

                </td>

            </tr>

        </table>

        <!-- Generic Pod Left Aligned with Price breakdown End -->

        <!-- Seperator Start -->

        <table cellpadding="0" cellspacing="0" cols="1" bgcolor="#d7d7d7" align="center" style="max-width: 600px; width: 100%;">

            <tr bgcolor="#d7d7d7">

                <td height="50" style="color: #464646; font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif; font-size: 14px; line-height: 16px; vertical-align:
```

```
top;">
                    </td>
                </tr>
            </table>
            <!-- Seperator End -->
        </td>
    </tr>
</table>
<!-- Content End -->
</body>
</html>
```

Declaration:

This is the page of your transaction details that you have filled the previous page that it will take it and gives the print of your transaction receipt that will help you that you have payed it or not from our website.

Result:

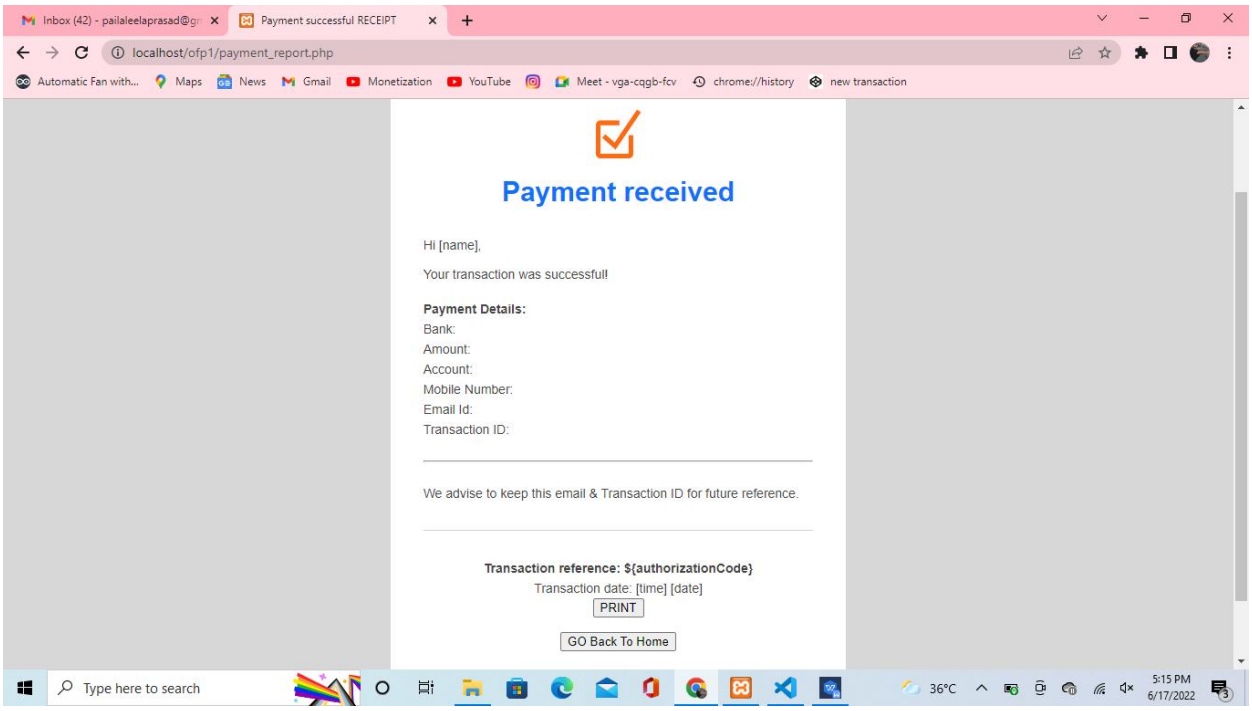

Figure 6.9: Payment Received Print

Testing

Chapter-7

Testing

7.1 System Testing and Validation Results.

System testing was done after the system was duly coded. Individual modules of the system were checked to ensure they are fully functional units before the integrating them. This was done by examining each unit; each script was checked to ensure that it functions as required and that it performed exactly as intended. The success of each individual unit gave us the go ahead to carryout integration testing. Different system modules were put together to make a complete system and integration testing ensured modules were compatible to be integrated to form a complete working system. The system was validated using a short questionnaire (Appendix C) that was filled by representatives of the users who were let to interact with the system using test data and provided feedback about the system features. This was done to assess if the system met their needs and requirements as regards paying fees to the university. It was found out that the system performed in conformance to the then defined user needs and requirements. Results of the validation are shown in Table 5.2 as percentages of respondents against each requirement.

Table 7.1: System testing and validation results

Requirement	Strongly Agree	Agree	Not Sure	Disagree	Strongly Disagree
System validates user input.	17%	78%	5%	0%	0%
System processes fees payment transactions.	17%	78%	5%	0%	0%
System provides payers with proof of payment for each transaction made.	20%	80%	0%	0%	0%
System provides guidelines on what is required to pay fees online.	10%	0%	70%	13%	7%
System generates reports of online transactions that are accessed by finance officers.	11%	76%	8%	5%	0%
System provides security to user data.	13%	70%	7%	5%	5%

Conclusion

Chapter 8
Conclusion

8.1 Conclusion

The project sought to develop an online fees payment system that provides relief of the long endured problems of the current modes of paying fees in PSCMRCET. Problems that students and their sponsors faced regarding paying fees to the university were identified and a solution was designed. Researchers developed a web based system that enables students and their sponsors to pay university fees from wherever they are using credit and debit cards. This system was welcomed by all its users who believed it would solve most of the problems and improve conditions regarding paying fees in PSCMRCET. The project achieved all its objectives and as a result, PSCMR-OFPS was designed, developed, tested and validated with real users. Hence, it was proved that this online system was fit to be implemented.

8.2 Limitations

i. Researchers did not involve all users in the project because of limited time; instead they worked with user representatives during data collection and system validation activities.
ii. The project involved the use of technical terms which required researchers to explicitly explain them during interactions with stakeholders that never knew the meanings of such terms.
iii. The researchers‟ attempts to access and connect to the university existing information systems were futile as managers and administrators of those systems claimed it would compromise their security.
iv. The developed product is a prototype and not a fully functional system that is integrated with other existing university systems.

8.3 Further Research

Refining the system developed by this project.
Establishing the feasibility of integrating mobile money payments into PSCMR-OFPS to enable fees payment by mobile money subscribers.

Providing a mobile based system with same functionality as PSCMR-OFPS to allow fees payment operations is done on mobile phones.

Bibliography

Chapter 9
Bibliography

- https://www.w3schools.com/html/html_intro.asp
- https://www.w3schools.com/css/default.asp
- https://www.w3schools.com/js/default.asp
- https://www.w3schools.com/sql/default.asp
- https://www.w3schools.com/php/default.asp
- https://www.studentstutorial.com/php/php-introduction