**ELECTRONIC ATTENDANCE AND SEATING ARRANGEMENT FOR UNIVERSITY EXAMINATIONS**

##### A DESIGN PROJECT REPORT

***Submitted by***

##### MUKESH SAI KUMAR M (17115007)

##### M BRAMAIAH (17113216)

##### P NIRANJAN BABU (17115003)

**Under the guidance of**

**Dr.KRISHNAVENI**

**&**

**Dr.S.SATHYALAKSHMI**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

****

**APRIL 2019**



**BONAFIDE CERTIFICATE**

Certified that this project report “**ELECTRONIC ATTENDANCE AND SEATING ARRANGEMENT FOR UNIVERSITY EXAMINATIONS”** is the bonafide work of**MUKESH SAI KUMAR M – 17115007, M BRAMAIAH – 17113216, P NIRANJAN BABU – 17115003** who carried out the project work under our supervision during the academic year 2018-2019.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
|  |  |
| Dr. S SATHYALAKSMI, Ph. D. | Dr. R. KRISHNAVENI, Ph. D. |
| **MENTOR** | **MENTOR** |
| Professor | Professor |
| Department of Computer Science&  Engineering | Department of Computer Science & Engineering |
|  |  |
| **INTERNAL EXAMINER** | **EXTERNAL EXAMINER** |
| Name: | Name: |
| Designation: | Designation: |

Project Viva - voce conducted on \_\_\_\_\_\_\_\_\_\_

**ACKNOWLEDGEMENT**

We thank **Dr. Rajeshwari Mukesh, Head of the Department of Computer Science & Engineering** for her strong support and encouragement throughout our course of study.

We express deep gratitude to our Mentors **Dr.S.Sathyalakshmi, Professor and Dr.S.Krishnaveni, Professor, Department of Computer Science & Engineering** for their constant guidance and continued assistance in completion of this design project.

We thank all the **Faculty members** and **Technical staff** of Department of Computer Science and engineering for their valuable support and suggestions at various stages of design project development.

We are extremely indebted to our family members for their adorable support and care.

***JEBERSON SAMUEL***

***SACHIN KAMAL***

***ASHWIN C***

**ABSTRACT**

Currently, most Universities in India put up attendance and seating arrangements in the form of paper notices in notice boards. This is very tedious and time-consuming both for the staff who put it up and for the students who must check it minutes before the exam. Thus we propose an electronic attendance and seating arrangement system for examinations. This will be implemented in the form of a web application for viewing and uploading the seating arrangement, along with an Android application for marking the attendance. The electronic system is very convenient for both staff and students as:

* Every student has access to Internet via her mobile phone
* Digitizing the records save paper
* Online systems are reliable and cannot be easily damaged/tampered with.

|  |
| --- |
|  |
|  |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | **iv** |
|  | **LIST OF FIGURES** | **vii** |
|  | **LIST OF ABBREVIATIONS** | **viii** |
| 1 | **INTRODUCTION** | **1** |
|  | 1.1 OVERVIEW | 1 |
|  | 1.2 MOTIVATION | 1 |
|  | 1.3 PROBLEM DEFINITION | 1 |
|  | 1.4 GOALS AND OBJECTIVES | 1 |
|  | 1.5 SCOPE AND APPLICATIONS | 1 |
| 2  3 | **LITERATURE REVIEW**  2.1 INTRODUCTION  2.2 SURVEY REGARDING EXISTING IMPLEMENTATIONS OF SIMILAR SYSTEMS  2.3 SUMMARY  PROJECT REQUIREMENT ANALYSIS  3.1 PROJECT REQUIREMENTS  3.1.1 SOFTWARE REQUIREMENTS  3.1.2 HARDWARE REQUIREMENTS  3.2 FEASIBILITY | **3**  3  3  5  5  5  5  5  5 |
| 4 | **SYSTEM DESIGN** | **6** |
|  | 4.1 SYSTEM ARCHITECTURE | 6 |
|  | 4.2 CROSS-FUNCTIONAL DIAGRAM  4.  4. 4.3 ENTITY RELATIONSHIP DIAGRAM  4. 4.4 METHODOLOGY | 7  9  10 |
| 5 | **IMPLEMENTATION**  5.1 OVERVIEW OF MySQL DATABASE  5.2 OVERVIEW OF PHP | **12**  12  12 |
| 6 | **CONCLUSION AND FUTURE WORK**  6.1 CONCLUSION  6.2 FUTURE WORK | **13**  13  13 |
|  | **REFERENCES** | **14** |
|  | **APPENDIX 1 - CODE SNIPPETS**  **APPENDIX 2 - SCREENSHOTS** | **15**  **23** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| 3.1 | SYSTEM INFRASTRUCTURE | 12 |
| 4.1 | SYSTEM ARCHITECTURE | 13 |
| 4.2 | CROSS FUNCTIONAL FLOW DIAGRAM | 14 |
| 4.3 | ENTITY RELATIONSHIP DIAGRAM | 15 |

**LIST OF SYMBOLS, ABBREVIATION & NOMENCLATURE**

|  |  |
| --- | --- |
| **ABBREVATION** | **TITLE** |
| PHP | HYPERTEXT PREPROCESSOR |
| SQL | STRUCTURED QUERY LANGUAGE |
| HTTP | HYPERTEXT TRANSFER PROTOCOL |
| Web app | WEB APPLICATION |

**CHAPTER -1**

**INTRODUCTION**

**1.1 OVERVIEW**

This project will be a platform for marking attendance and displaying seating arrangement to the students in a convenient manner, during the University examinations. This is being developed mainly as a convenience and as a more stable, fast, easily reachable and reliable alternative to the existing system. This also saves the time of all parties involved, as there is no need to put up paper notices, if the proposed system is implemented properly. There is also no possibility of record tampering, as the staff is responsible for marking the attendance of the student. Thus, no student can sign in place of another, proving the security of the proposed system.

**1.2 MOTIVATION**

We, as students, have experienced several hardships when checking for our seating arrangements. For instance, we had our roll numbers under two different rooms, so we lost a lot of examination time finding the correct room. Getting rid of this unnecessary delay was the primary motivation of this project. Electronic records have the added advantage of being persistent and very difficult to tamper with. Thus, this system is also secure, stable and reliable.

**1.3 PROBLEM DEFINITION**

An application must be created wherein staff are able to upload seating arrangement for examinations and a student is able to view her seating arrangement. This application must also be capable of holding the attendance records of students during examinations.

**1.4 GOALS AND OBJECTIVES**

* To provide a stable electronic data storage system.
* To secure the system against unauthorized access.
* To make the system easily available for all parties involved.
* To simplify the process of conducting an examination.

**1.5 SCOPE AND APPLICATIONS**

This project can be implemented as a real, working product in any University for managing their attendance and seating arrangements. This can also be used by other institutions such as schools, boards of education and government institutions to simplify the process of conducting their examinations.

The logic of this product can also be extended to provide dynamic reservation systems such as those for buses and trains. For example, users can reserve seats for themselves after viewing the available seats. If the reserved seats are not taken, it can be allocated to other users after confirming that the seat will remain empty.

**CHAPTER -2**

**LITERATURE REVIEW**

**2.1 INTRODUCTION**

This is an online application for marking seating arrangement through a web application. This system can also be used by the staff for marking attendance of the students. This online system is more reliable than the traditional using system.

**2.2 SURVEY REGARDING EXISTING IMPLEMENTATIONS OF SIMILAR SYSTEMS**

We carried out surveys about similar systems being implemented in colleges across Chennai, Andhra, Punjab and Telangana.

**Chennai:**

**Bharath University:**

* For attendance they are using android app.
* For seating arrangement of university exam, they provide manually.

**Satyabama University:**

* For attendance they are taking manually.
* For seating arrangement of university exam, they provide manually.

**SRM University:**

* For attendance they are providing manually.
* For seating arrangement of university exam, they are using website.

**Andhra:**

**Sri Venkateswara University:**

* For attendance they are providing manually.
* For seating arrangement of university exam, they are providing manually.

**Telangana:**

**Osmania University:**

* For both they are using manual process.

**Punjab:**

**LPU:**

* For both they are using android application and college website.

**2. SUMMARY**

The above paper discusses the previous work done by the other researchers in the field of online seating arrangement, attendance and implementation. Each paper attempts to solve different aspects of the problem by combining and implementing new algorithms etc.,

**CHAPTER -3**

**3. PROJECT REQUIREMENT ANALYSIS**

**3.1 PROJECT REQUIREMENT**

**3.1.1 Software Requirement**

* + Programming Language: PHP, JavaScript, SQL
  + Front–End: HTML, CSS
  + Database: MySQL Database
  + IDE: Sublime Text Editor
  + Web Server: Apache HTTP Server
  + OS: Linux

**3.1.2 Hardware Requirement**

* + Processor: Intel® Core II Duo
  + RAM: 2 GB
  + HDD: 10 GB
  + Display: Color Monitor

**3.2 FEASIBILITY STUDY**

The storage space required to store the data of about a few thousand students will be in the order of a few gigabytes and this is easily satisfied by a modern personal computer. The data will be stored in a DBMS thus facilitating efficient, reliable and hard-to-corrupt storage. The Web App will act as an interface to the database and will allow viewing of the data stored in the database and updating the relevant information. The Apache HTTP Server and PHP Scripting engine along with MySQL DBMS will require about a gigabyte of primary memory to run smoothly. Thus the storage, processing power and memory required is easily satisfied by even a personal computer, which makes this project really cheap and affordable to implement whilst being handy and time-saving.

**CHAPTER -4**

**SYSTEM DESIGN AND METHODOLOGY**

[**4.1 SYSTEM ARCHITECTURE**](#_30j0zll)

Describe the system architecture, or simply provide the architecture diagram. For Fyndit Application it may include android based front end, and database server (Three-tier architecture).

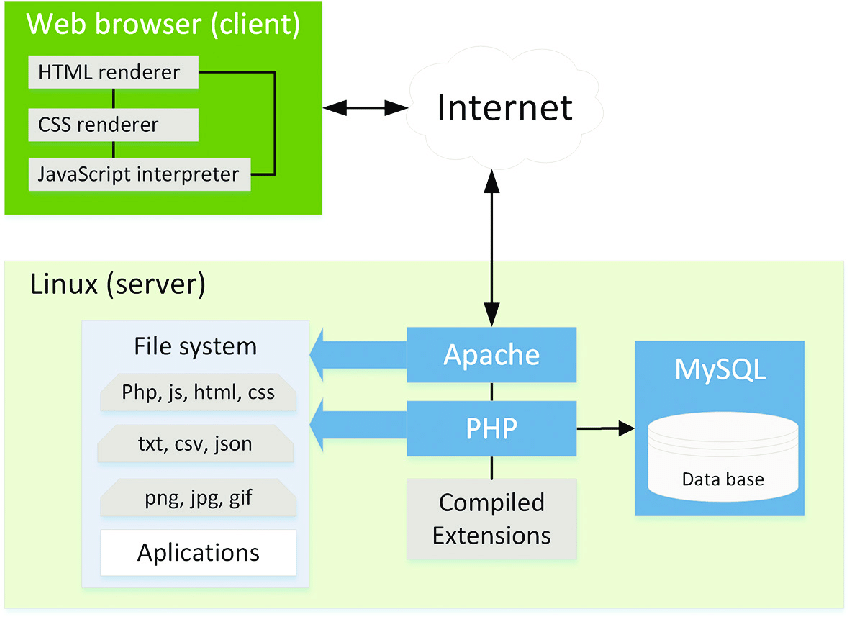


Fig 4.1: System Architecture

**4.2 CROSS-FUNCTIONAL FLOW DIAGRAM (WORKFLOW)**

A cross-functional diagram is referred to as deployment flowchart or swim lane flowchart, is a type of process mapping flowchart. This type of diagrams illustrates the relationship between stakeholders within a process. This is achieved via the usage of columns or rows generally referred to as lanes. Each person, team, or department that is involved with the process is represented through a lane where are listed the activities of the process it is responsible for.

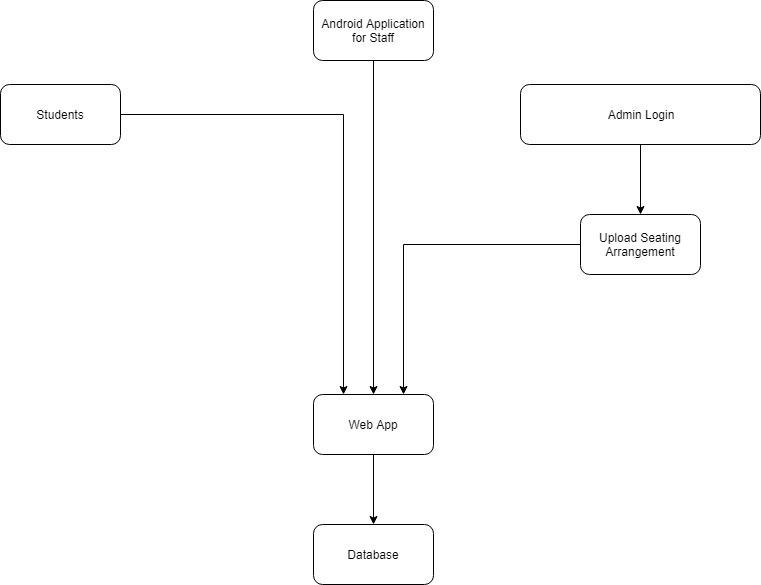


Fig 4.2: Workflow

**4.3 ENTITY-RELATIONSHIP DIAGRAM**

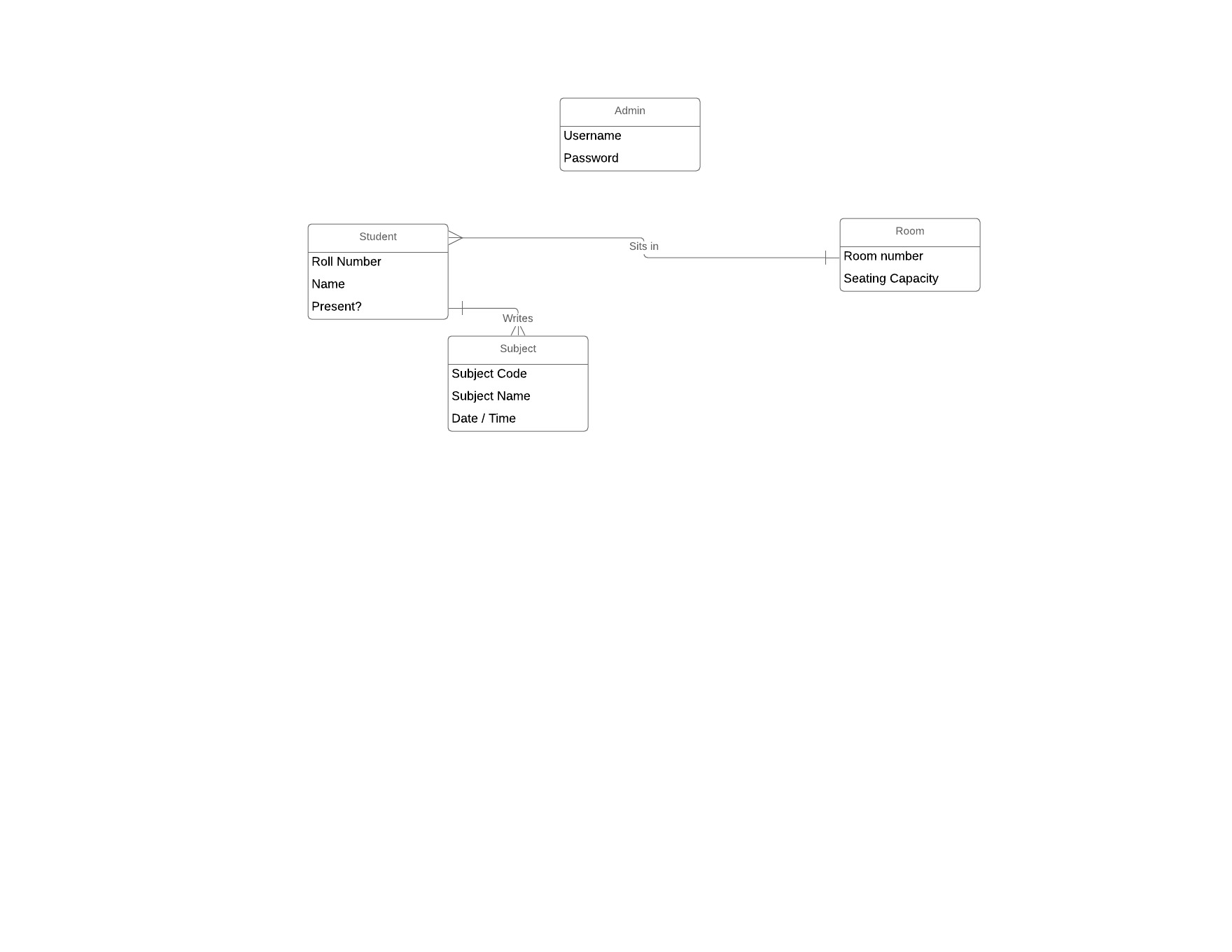


Fig 4.3: ER Diagram

**4.4 METHODOLOGY**

The PHP scripts written are deployed into the document root of the server. The HTTP Server is configured to handle PHP files through a PHP handler which executes the scripts. The database is configured with the required tables. A database user is given access to the database and is used to retrieve and store values. Then the MySQL database and web servers are linked by installing the required support packages for the web server. The entire application is tested by executing all functionality inside the application. If any errors are encountered, the required fixes in the application/server are made and the application is tested this way until it is ready for a stable release. The application is also tested on different platforms and clients to ensure that it is compatible with the maximum number of devices. After the final release, the application is made public by exposing the HTTP Server to the Internet. By this time, all bugs in the application are fixed and it is working as intended.

**CHAPTER -5**

**IMPLEMENTATION**

# 5.1 OVERVIEW OF MySQL DATABASE

MySQL is a relational database management system (RDBMS) released under the GNU General Public License (GPL). It is one of the most widely used open-source database systems and is compatible with a multitude of website applications.

In MySQL, the storage engine specifies how the database contents are actually stored physically on the computer running the DBMS.

 Here is a list of the available storage engines:

mysql> show engines;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Engine** | **Support** | **Comment** | **Transactions** | **XA** | **Savepoints** |
| CSV | YES | CSV storage engine | NO | NO | NO |
| MRG\_MYISAM | YES | Collection of identical MyISAM tables | NO | NO | NO |
| FEDERATED | NO | Federated MySQL storage engine | NULL | NULL | NULL |
| MyISAM | DEFAULT | Default engine as of MySQL 3.23 with great performance | NO | NO | NO |
| InnoDB | YES | Supports transactions, row-level locking, and foreign keys | YES | YES | YES |
| MEMORY | YES | Hash based, stored in memory, useful for temporary tables | NO | NO | NO |

**5.2 OVERVIEW OF PHP**

**What Exactly is PHP?**

PHP is an intuitive, server-side scripting language. Like any other scripting language, it allows developers to build logic into the creation of web page content and handle data returned from a web browser. PHP also contains several extensions that make it easy to interact with databases, extracting data to be displayed on a web page and storing information entered by a web site visitor back into the database.

PHP consists of a scripting language and an interpreter. Like other scripting languages, PHP enables web developers to define the behavior and logic they need in a web page. These scripts are embedded into the HTML documents that are served by the web server. The interpreter takes the form of a module that integrates into the web server, converting the scripts into commands the computer then executes to achieve the results defined in the script by the web developer.

**How Does PHP Work?**

To develop an understanding of how PHP works it is helpful to first explore what happens when a web page is served to a user's browser.

When a user visits a web site or clicks on a link on a page the browser sends a request to the web server hosting the site asking for a copy of the web page. The web server receives the request, finds the corresponding web page file on the file system and sends it back, over the internet, to the user's browser.

Typically, the web server doesn't pay any attention to the content of the file it has just transmitted to the web browser. As far as the web server is concerned the web browser understands the content of the web page file and knows how to interpret and render it so that it appears as the web designer intended.

Now let's consider what kind of web page content a web browser understands. These days a web page is likely to consist of HTML, XHTML and JavaScript. The web browser contains code that tells it what to do with these types of content. For example, it understands the structure HTML in terms of rendering the page and it has a JavaScript interpreter built in that knows how to execute the instructions in a JavaScript script. A web browser, however, knows absolutely nothing about any PHP script that may be embedded in an HTML document. If a browser was served a web page containing PHP it would not know how to interpret that code.

Given that a web browser knows nothing about PHP in a web page, then clearly something has to be done with any PHP script in the page before it reaches the browser. This is where the PHP pre-processing module comes in. The PHP module is, as mentioned previously, integrated into the web server. The module tells the web server that when a page is to be served which contains PHP script (identified by special markers) that it is to pass that script to the PHP pre-processing module and wait for the PHP module to send it some content to replace that script fragment. The PHP processing module understands PHP, executes the PHP script written by the web developer and, based on the script instructions, creates output that the browser will understand. The web server substitutes the content provided by the PHP pre-processor module in place of the PHP script in the web page and sends it to the browser where it is rendered for the user to view.

**CHAPTER -6**

**CONCLUSION & FUTURE ENHANCEMENT**

**6.1 CONCLUSION**

Web applications is not just a change of technology. It is part of a redefinition of how work can be efficiently done without having to waste time on mundane, repetitive and easily automatable tasks. I will end this book by daring to make a few predictions of how this application and the functions it serves will continue to develop.

CONDUCT AN EXAM EASILY: It can be used by any institution which conducts examinations to get rid of the burden of conducting examinations and focus more on the quality of examination.

MANAGEMENT APPLICATIONS: The logic of this application can be extended to provide applications which can manage any given resource which must be available to several people at once.

**6.2 FUTURE WORK**

* Feedback column will be added for better user experience
* User interface will be more precise and eloquent
* Android application will be developed to mark attendance

**CHAPTER -7**

**REFERENCES**

* Ken Kousen published Gradle Recipes for android was originally released
* in 2016
* Ben Smith published Beginning JSON - JavaScript object notation was originally released in 2014
* [Reto Meier](https://www.google.com/search?sa=X&rlz=1C1CHBD_enIN842IN842&biw=1920&bih=969&q=Reto+Meier&stick=H4sIAAAAAAAAAOPgE-LVT9c3NEypyC7PSTbMUeIBcY3yKlJMLS1MtWSyk630k_Lzs_XLizJLSlLz4svzi7KtEktLMvKLFrFyBaWW5Cv4pmamFgEAw16NbUoAAAA&ved=2ahUKEwiVzonFidThAhUNUI8KHZMNAp4QmxMoATAQegQIDxAK), [Ian Lake](https://www.google.com/search?sa=X&rlz=1C1CHBD_enIN842IN842&biw=1920&bih=969&q=Ian+Lake&stick=H4sIAAAAAAAAAOPgE-LVT9c3NEypyC7PSTbMUYJw07NyLQyyTIq1ZLKTrfST8vOz9cuLMktKUvPiy_OLsq0SS0sy8osWsXJ4JuYp-CRmpwIAf0AMtUkAAAA&ved=2ahUKEwiVzonFidThAhUNUI8KHZMNAp4QmxMoAjAQegQIDxAL) Professional android used for jdk, sdk, java problem was originally released in 2018 by
* Jessica Thornsby published Android UI design with xml published in 2016 may
* [https://stackoverflow.com](https://visualstudio.microsoft.com/)
* [http://www.developers.google.com](http://www.ajaxcontroltoolkit.com/)
* [https://www.firebase.google.com](https://www.tutorialspoint.com/javascript/)
* [https://www.devslopes.com](https://www.tutorialspoint.com/mysql/mysql-introduction.htm)

**APPENDIX 1**

**CODE SNIPPETS**

**Mobile readable code(XML):**

***<?*xml version="1.0" encoding="utf-8"*?>***

**<manifest xmlns:android="http://schemas.android.com/apk/res/android"**

**package="com.example.fyndit">**

**<uses-permission android:name="android.permission.INTERNET" />**

**<application**

**android:allowBackup="true"**

**android:icon="@mipmap/ic\_launcher"**

**android:label="@string/app\_name"**

**android:roundIcon="@mipmap/ic\_launcher\_round"**

**android:supportsRtl="true"**

**android:theme="@style/AppTheme">**

**<activity android:name=".course"></activity>**

**<activity android:name=".signin\_Activity" />**

**<activity android:name=".signup\_Activity" />**

**<activity**

**android:name=".MainActivity"**

**android:label="@string/app\_name"**

**android:theme="@style/AppTheme.NoActionBar">**

**<intent-filter>**

**<action android:name="android.intent.action.MAIN" />**

**<category android:name="android.intent.category.LAUNCHER" />**

**</intent-filter>**

**</activity>**

**</application>**

**</manifest>**

From the above code it’s clear that there are four main activities

* Main\_activity
* sign\_in
* sign\_up
* course

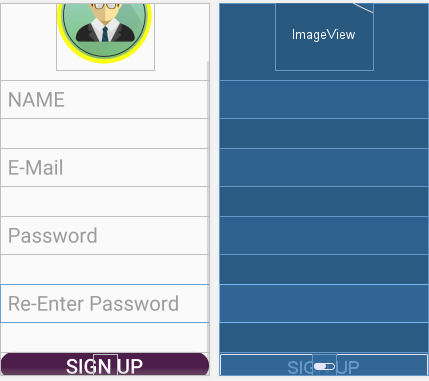
This code also called as manifest file which Runs when a device is yet to launch an app

**APPENDIX 2**

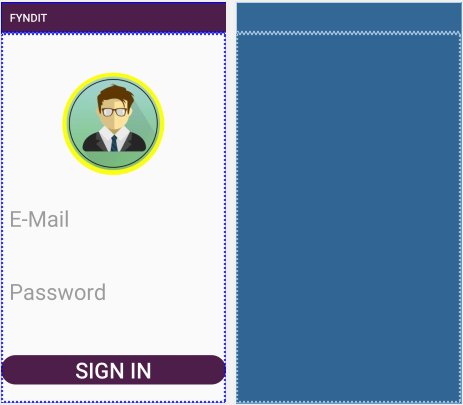
**SAMPLE SCREEN SHOTS: OUTPUT**



* The above given figure is the home page of the application



* The above given figure is the depiction of the sign up page of the application



* The above given figure is the depiction of the sign in page of the application

