A Project Report
on

IOT BASED ATTENDENCE SYSTEM USING BLOCKCHAIN

submitted in partial fulfillment of the requirements for the award of the degree
of
BACHELOR OF TECHNOLOGY
in
COMPUTER SICENCE AND ENGINEERING

by

17WH1A05B6	**Ms. G.MANASA**
17WH1A05A4	**Ms. M.SRILATHA**
18WH5A0513	**Ms. L.SOUMYA**

under the esteemed guidance of
Mr. M.BAPIRAJU
Assistant Professor

Department of Computer Science and Engineering
BVRIT HYDERABAD
College of Engineering for Women
(NBA Accredited – EEE, ECE, CSE and IT)
(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)
Bachupally, Hyderabad – 500090

June, 2021

DECLARATION

We hereby declare that the work presented in this project entitled **"IOT BASED ATTENDENCE SYSTEM USING BLOCKCHAIN"** submitted towards completion of Project Work in IV year of B.Tech., CSE at 'BVRIT HYDERABAD College of Engineering For Women', Hyderabad is an authentic record of our original work carried out under the guidance of Mr. M.BapiRaju, Assistant Professor, Department of CSE.

Sign. with date:
Ms. G.MANASA
(17WH1A05B6)

Sign. with date:
Ms. M.SRILATHA
(17WH1A05A4)

Sign. with date:
Ms. L.SOUMYA
(18WH5A0513)

BVRIT HYDERABAD
College of Engineering for Women
(NBA Accredited – EEE, ECE, CSE and IT)
(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)
Bachupally, Hyderabad – 500090

Department of Computer Science and Engineering

Certificate

This is to certify that the Project Work report on "**IOT BASED ATTENDENCE SYSTEM USING BLOCKCHAIN**" is a bonafide work carried out by Ms. G.MANASA (17WH1A05B6) ; Ms. M.SRILATHA (17WH1A05A4) ; Ms. L.SOWMYA (18WH5A0513) in the partial fulfillment for the award of B.Tech. degree in **Computer Science and Engineering, BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad**, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad under my guidance and supervision.

The results embodied in the project work have not been submitted to any other University or Institute for the award of any degree or diploma.

Head of the Department
Dr. K.Srinivasa Reddy
Professor and HoD,
Department of CSE

Guide
Mr.M.BapiRaju
Assistant Professor

External Examiner

Acknowledgements

We would like to express our sincere thanks to **Dr. K V N Sunitha, Principal**, **BVRIT HYDERABAD College of Engineering for Women**, for providing the working facilities in the college.

Our sincere thanks and gratitude to our **Dr. K.Srinivasa Reddy, Professor**, Department of CSE, **BVRIT HYDERABAD College of Engineering for Women** for all the timely support and valuable suggestions during the period of our project.

We are extremely thankful and indebted to our internal guide, **Mr. M.BapiRaju, Assistant Professor**, Department of CSE**, BVRIT HYDERABAD College of Engineering for Women** for his constant guidance, encouragement and moral support throughout the project.

Finally, we would also like to thank our Project Coordinator, all the faculty and staff of **CSE** Department who helped us directly or indirectly, parents and friends for their cooperation in completing the project work.

Ms. G.MANASA

(17WH1A05B6)

Ms. M.SRILATHA

(17WH1A05A4)

Ms. L.SOUMYA

(18WH5A0513)

Contents

ABSTRACT

Traditionally, student ‘s attendances are taken manually by using attendance sheet given by the faculty members in class, which is a time-consuming event. Moreover, it is very difficult to verify one by one student in a large classroom environment with distributed branches whether the authenticated students are actually responding or not. Using conventional method of calling out names takes approximately 5-10 minutes for marking attendance of entire class. It becomes complicated when strength is more. With the increase in technology, attendance monitoring is designed with android or web-based applications. However, the intention of this design is to provide a Blockchain based app that can be downloaded and used by the organization with no third-party control to meddle with the data. On a blockchain based system, no administrator permission is allowed to editing or deleting data. Someone who inserts an information record on the blockchain will not be able to deny that he is doing the activity. There is an update option to modify attendance when it’s needed. However, the modifications are recorded and tracked, just in case it’s a fraudulent activity. Attendance is captured using IOT automatically and is entered into the blockchain which makes the data tamper-proof, secure and robust. The privacy of it’s users is preserved because the user ids are generated by trusted third party. This data is available for government for Scholarship and other related decision making.

LIST OF FIGURES

1. INTRODUCTION

IOT based Attendance System using Blockchain is an application that is made for students and faculty of a particular college to maintain students' attendance which is captured through an IOT device(biometric) and then the attendance is stored in the Blockchain. Blockchain is used in this application to ensure the safety and tamper-free environment as the data cannot be manipulated and is used for government purposes.

1.1 Objectives

Generally, in many institutions attendance is monitored and marked using conventional systems like android or other similar web applications. Few conventional databases do not have features like checking whether any information has experienced unauthorized changes or not. In this system when the data is entered into the blockchain, no one is allowed to edit or delete the data. This makes the application transparent and different from other web-based attendance systems as IOT is used to capture the attendance through biometric of the students in the class. Students' poor attendance rate is one of the most challenging problems tackled by the college management today. With the help of this application student's attendance rate can be improved which is also helpful for government to take precise decisions regarding scholarship like schemes for students with transparent data. Using blockchain and some encryption techniques, this application is made secure from any manipulations.

1.2 Methodology

The fingerprint module will collect the fingerprint data from the multiple users and sends it over the internet to the website. The Enrolment of fingerprints is done on the Server and verification is done on the client with the transmission of fingerprint templates over the network. The website is coded in HTML, CSS, JSP has a MySQL database and records of attendance stored in Blockchain. By logging into the website, the student can view all their attendance records. The timestamp of students' attendance is encrypted and stored in the blockchain.

1.2.1 Introduction

Normally in schools and colleges, the average student ‘s count will be 50-80. Faculty has to mark student ‘s presence for every hour. Traditionally, student ‘s attendances are taken manually by using attendance sheet given by the faculty members in class, which is a time-consuming event. Moreover, it is very difficult to verify one by one student in a large classroom environment with distributed branches whether the authenticated students are actually responding or not. Using conventional method of calling out names takes approximately 5-10 minutes for marking attendance of entire class. It becomes complicated when strength is more. This application is designed to overcome the conventional system also considering the security of the data as with the increase in technology, the attacks on applications are also widely increased. Attackers are trying to hack websites to modify content. Blockchain technology is being introduced to make these attacks impossible.

1.2.2 IOT and Biometrics

IOT is best served by a set of secure data points; it relies on the integrity of the data sent and received. Those data points share vital information and make important connections that establish relationships and recommendations. Those recommendations often contain sensitive user data–this is where the security of biometrics becomes most important and a key player in strong security for connected devices and retained data. Biometrics provide a secure way to transfer data as well as identify data ports and devices and ensure that they remain secure and their data intact. Biometrics are an optimal security measure, and their continued development will be a key component to creating difficult to breach security protocols. Since the characteristics identified by biometric scanners do not change and are unique to each individual, they make a very secure means of communicating data and creating identifiers for sharing secured data.

Fig 1.2.1: IOT Fingerprint Module

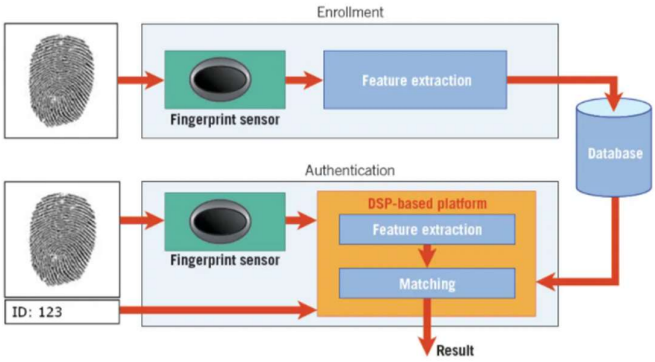

Fig 1.2.2: Block Diagram of fingerprint processing

1.2.3 Blockchain Technology

A blockchain is a growing list of records, called blocks, that are linked together using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree).

Blocks contain the hash of the previous block, forming a chain, with each additional block reinforcing the ones before it. Therefore, blockchains are resistant to modification of their data because once recorded, the data in any given block cannot be altered retroactively without altering all subsequent blocks.

The blockchain was invented by a person (or group of people) using the name Satoshi Nakamoto in 2008 to serve as the public transaction ledger of the cryptocurrency bitcoin.

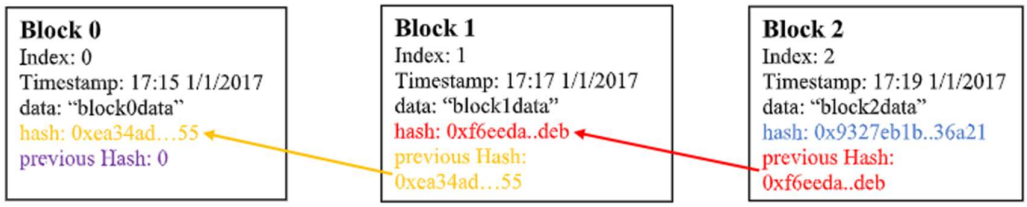

Fig 1.2.3: Blockchain

1.3 Organization of Project

The development of the project is done in a structured way.

We have three phases in our project.

- Project creation
- Database creation
- Developing front-end
- Connecting front-end to back-end
- Implementing blockchain

2. THEORETICAL ANALYSIS OF THE PROPOSED PROJECT

2.1 Requirements Gathering

2.1.1 Software Requirements

Programming Language : Java

Graphical User Interface: HTML, CSS with Bootstrap, JSP

Libraries : MYSQL connector jar file, Apache Tomcat jar file

Encryption Algorithm : SHA-256

API : JDBC

Framework : Java EE

Tool : Eclipse, MYSQL

2.1.2 Hardware Requirements

Operating System: Windows 10

Processor : Intel Core i5

CPU Speed : 2.30 GHz

Memory : 4 GB (RAM)

2.2 Technologies Description

Java

Python Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages.

The most important features of Java language are Simple, Object-Oriented, Portable, Platform independent, Secured, Robust, Architecture neutral, Interpreted, High Performance, Multithreaded, Distributed, Dynamic

HTML

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages.

- Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
- As its name suggests, HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, and variations in display for different devices and screen sizes as well as a variety of other effects.

JSP

Java Server Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications. JSP have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. JSP is one of the most widely used language over the web. A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

SHA-256 Encryption Algorithm

SHA-256 is one of the successor hash functions to SHA-1 (collectively referred to as SHA-2), and is one of the strongest hash functions available. SHA-256 is not much more complex to code than SHA-1, and has not yet been compromised in any way. The 256-bit key makes it a good partner-function for AES. It is defined in the NIST (National Institute of Standards and Technology) standard 'FIPS 180-4'. NIST also provide a number of test vectors to verify correctness of implementation.SHA-256, messages up to 2^{64} bit (2.3 exabytes, or 2.3 billion gigabytes) are transformed into digests of size 256 bits (32 bytes). For perspective, this means that an object 7 times the size of Facebook's data warehouse in 2014 passed to SHA-256 would produce a chunk of data the size of a 32-letter string of ASCII characters, and that string would the object's very special fingerprint.

JDBC

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database, and is oriented toward relational databases.

Java EE Framework

The Java EE stands for Java Enterprise Edition, which was earlier known as J2EE and is currently known as Jakarta EE. It is a set of specifications wrapping around Java SE (Standard Edition). The Java EE provides a platform for developers with enterprise features such as distributed computing and web services. Java EE applications are usually run-on reference run times such as micro servers or application servers. Examples of some contexts where Java EE is used are e-commerce, accounting, banking information systems. Java EE has several specifications which are useful in making web pages, reading and writing from database in a transactional way, managing distributed queues. The Java EE contains several APIs which have the functionalities

of base Java SE APIs such as Enterprise JavaBeans, connectors, Servlets, Java Server Pages and several web service technologies.

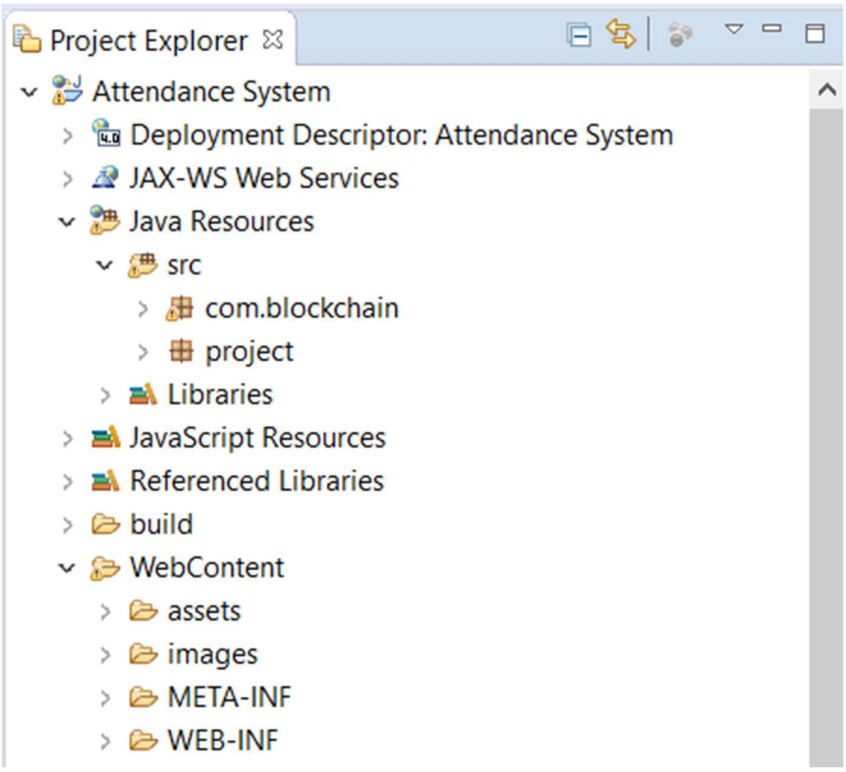

Fig 2.2: Java EE project structure

MYSQL Connector jar file

MySQL Connector/J is the official JDBC driver for MySQL. MySQL jar file has to be downloaded from the official website. This jar file would help to connect with the database. JDBC API should be used to connect to the database. JDBC consists of the code for this connection.

Eclipse

Eclipse is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins

The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

MySQL

MySQL is an open-source relational database management system (RDBMS). A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmer 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).

3. DESIGN

3.1 Introduction

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement has been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

3.2 Architecture Diagram

Web applications are by nature distributed applications, meaning that they are programs that run on more than one computer and communicate through network or server. Specifically, web applications are accessed with a web browser and are popular because of the ease of using the browser as a user client. For the enterprise, software on potentially thousands of client computers is a key reason for their popularity. Web applications are used for web mail, online retail sales, discussion boards, weblogs, online banking, and more. One web application can be accessed and used by millions of people.

Like desktop applications, web applications are made up of many parts and often contain mini programs and some of which have user interfaces. In addition, web applications frequently require an additional markup or scripting language, such as HTML, CSS, or JavaScript programming language.

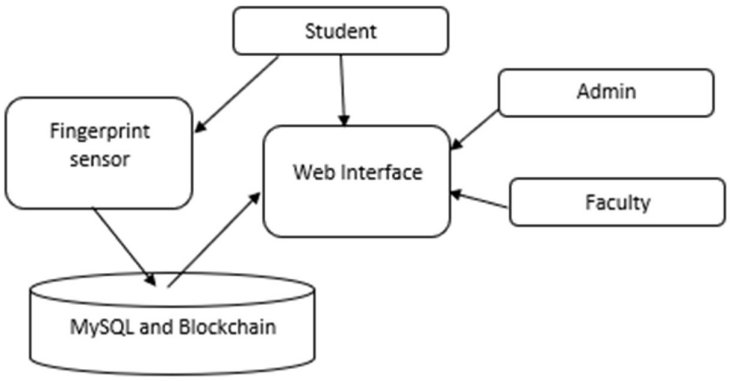

Fig 3.2: Architecture Diagram

3.3 UML Diagrams

3.3.1 Use Case Diagram

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating.

Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consist of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

Hence to model the entire system, a number of use case diagrams are used.

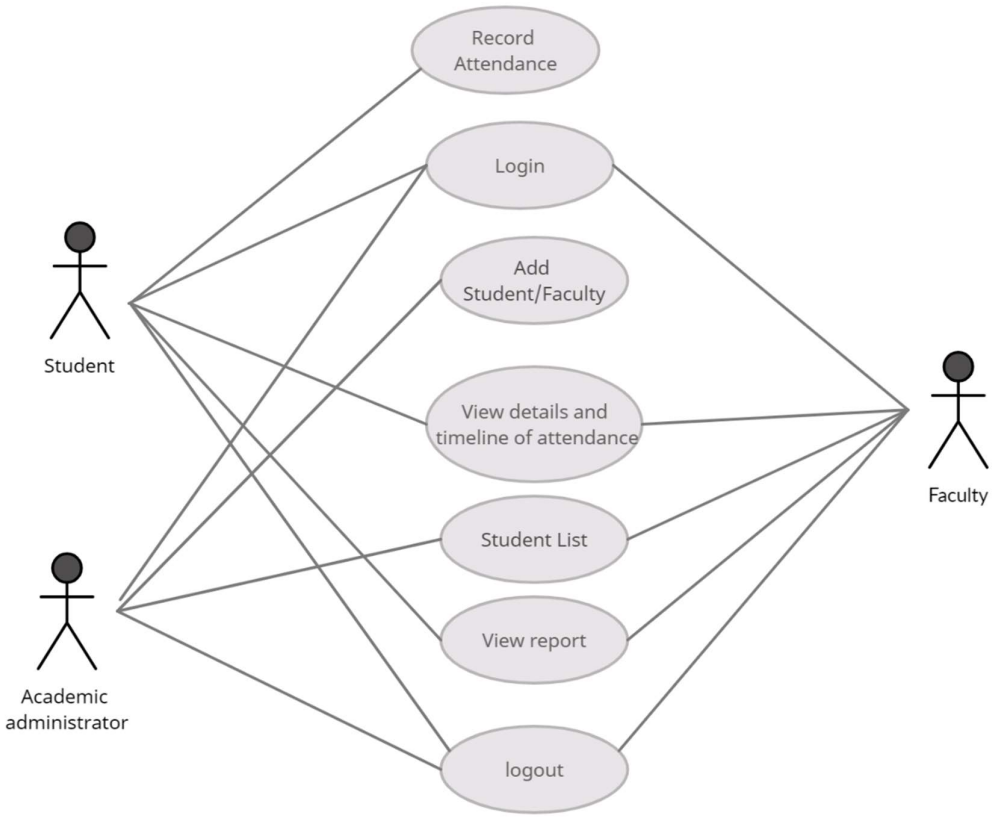

Fig 3.3.1: Use Case Diagram

3.3.2 Sequence Diagram

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

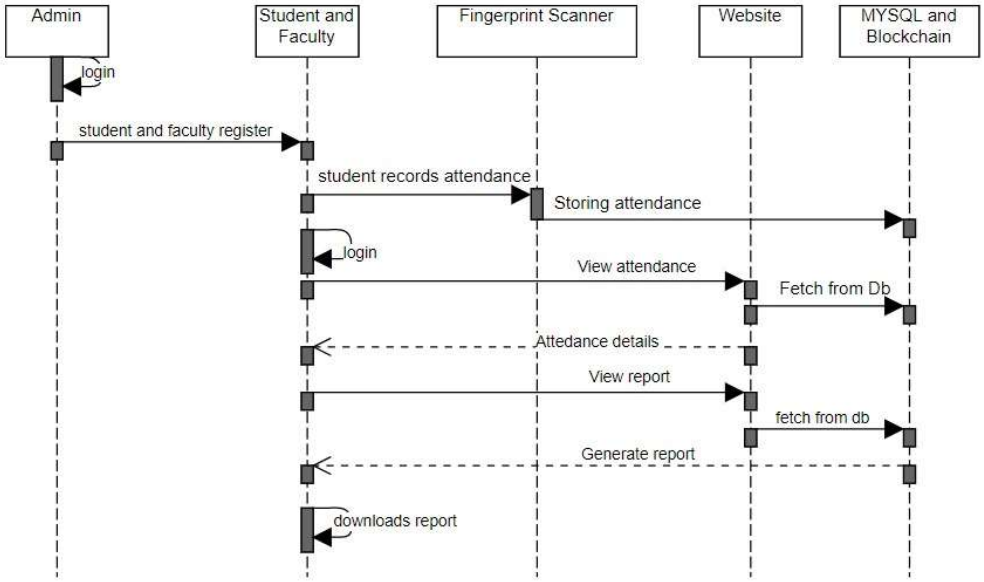

Fig 3.3.2: Sequence Diagram

3.3.3 Activity Diagram

Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

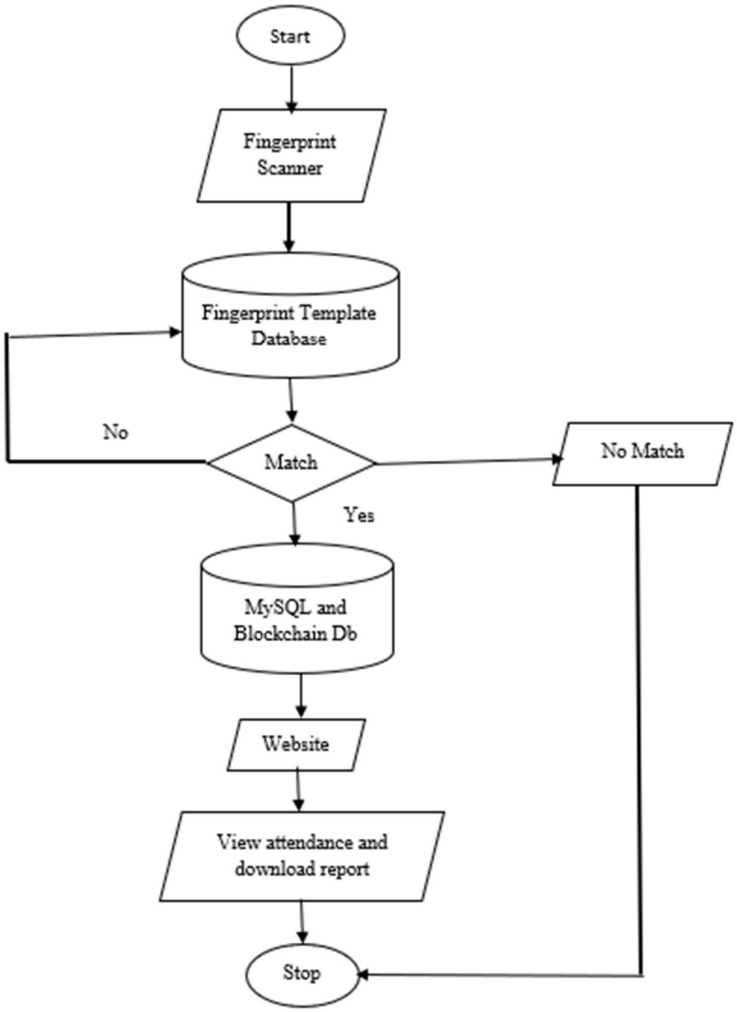

Fig 3.2.3: Activity Diagram

4. IMPLEMENTATION

4.1 Coding

index.html

```
<!DOCTYPE HTML>
<html>
<head>
<title>Attendance System</title>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<link rel="stylesheet" href="assets/css/main.css" />
</head>
<body>
<header id="header">
<div class="inner">
<a href="index.html" class="logo"><strong>BVRITH</strong></a>
<nav id="nav">
<a href="StudentLogin.html">Student</a>
<a href="adminLogin.html">Admin</a>
<a href="facultyLogin.html">Faculty</a>
</nav>
<a href="#navPanel" class="navPanelToggle"><span class="fa fa-bars"></span></a>
</div>
</header>
<section id="banner">
<div class="inner">
<header>
<h1>BVRITH Attendance Portal</h1>
</header>
<div class="flex ">
<div>
<h3>Address</h3>
<p>8-5/4 Bachupally, Opp:Rajiv Gandhi Nagar Colony,</p>
<p>Nizampet Rd, Hyderabad, Telangana 500090</p>
```

```
</div>
<div>
<h3>Contact Number</h3>
<p>040 4241 7773</p>
</div>
<div>
<h3>Email Address</h3>
<p>bvrithyderabad@gmail.com</p>
</div>
</div>
</div>
</section>
<script src="assets/js/jquery.min.js"></script>
<script src="assets/js/skel.min.js"></script>
<script src="assets/js/util.js"></script>
<script src="assets/js/main.js"></script>
 <hr class="new1">
</body>
</html>
```

addFaculty.jsp

```
<%@page import="project.ConnectionProvider" %>
<%@page import="java.sql.*"%>
<%@page import="java.io.FileInputStream"%>
<%@page import = "java.io.InputStream"%>
<%@page import="java.io.File"%>

<%
        String empId = request.getParameter("empId");
        String empName = request.getParameter("empName");
        String department = request.getParameter("department");
        String mobile = request.getParameter("mobile");
        String designation = request.getParameter("designation");
```

```
        try{
                Connection con = ConnectionProvider.getCon();
                System.out.println("Connection established.....");

                Statement st = con.createStatement();
                st.executeUpdate("insert into faculty values('"+empId+"',
'"+empName+"', '"+department+"', '"+mobile+"', '"+designation+"')");

            System.out.println("Record inserted.....");
            response.sendRedirect("adminHome.jsp");
          }catch (Exception e) {
       System.out.println(e);
    }
%>
```

addNewStudent.jsp

```
<%@page import="project.ConnectionProvider" %>
<%@page import="java.sql.*"%>
<%@page import="java.io.FileInputStream"%>
<%@page import = "java.io.InputStream"%>
<%@page import="java.io.File"%>

<%
        String course = request.getParameter("course");
        String branch = request.getParameter("branch");
        String rollNo = request.getParameter("rollNo");
        String name = request.getParameter("name");
        String fatherName = request.getParameter("fatherName");
        String mobile = request.getParameter("mobile");

        try{

                Connection con = ConnectionProvider.getCon();
                System.out.println("Connection established.....");

                Statement st = con.createStatement();
                st.executeUpdate("insert into student values('"+course+"', '"+branch+"',
'"+rollNo+"', '"+name+"', '"+fatherName+"', '"+mobile+"')");

            System.out.println("Record inserted.....");
```

```
            response.sendRedirect("adminHome.jsp");
          }catch (Exception e) {
        System.out.println(e);
    }
%>
```

adminHome.jsp

```
<%@include file="header.html"%>
<!DOCTYPE html>
<html>
<title>Admin</title>
<body>
<img src="./images/vishnuLogo.png"  align="left"width="100" height="100">

  <div class="w3-bar w3-black">
    <button class="w3-bar-item w3-button tablink w3-red"
onclick="openCity(event,'London')">Add New Student</button>
    <button class="w3-bar-item w3-button tablink"
onclick="openCity(event,'Paris')">Add Faculty</button>
    <button class="w3-bar-item w3-button tablink"
onclick="openCity(event,'Tokyo')">Student List</button>
     <button class="w3-bar-item w3-button tablink"
onclick="openCity(event,'Tokyo1')">Student Time Records</button>
     <a href="index.html" class="w3-bar-item w3-button tablink">Logout</a>
  </div>

  <div id="London" class="w3-container w3-border city">
  <br>
   <link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
<script
src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <link rel="dns-prefetch" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css?family=Raleway:300,400,600"
rel="stylesheet" type="text/css">
    <link rel="icon" href="Favicon.png">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
```

```
</head>
<body>
<main class="my-form">
   <div class="cotainer">
      <div class="row justify-content-center">
         <div class="col-md-8">
               <div class="card">
                  <div class="card-header">Add New Students</div>
                  <div class="card-body">
                     <form name="my-form" onsubmit="return validform()"
action="addNewStudent.jsp" method="post">
                        <div class="form-group row">
                           <label for="full_name" class="col-md-4 col-form-label text-
md-right">Course Name</label>
                           <div class="col-md-6">
                              <input type="text"  class="form-control" name="course">
                           </div>
                        </div>

                        <div class="form-group row">
                           <label for="email_address" class="col-md-4 col-form-label
text-md-right">Branch Name</label>
                           <div class="col-md-6">
                              <input type="text"  class="form-control" name="branch">
                           </div>
                        </div>

                        <div class="form-group row">
                           <label for="user_name" class="col-md-4 col-form-label text-
md-right">Roll Number</label>
                           <div class="col-md-6">
                              <input type="text" class="form-control" name="rollNo">
                           </div>
                        </div>

                        <div class="form-group row">
                           <label for="phone_number" class="col-md-4 col-form-label
text-md-right">Name</label>
                           <div class="col-md-6">
                              <input type="text"  class="form-control" name="name">
                           </div>
                        </div>

                        <div class="form-group row">
                           <label for="present_address" class="col-md-4 col-form-label
text-md-right">Father Name</label>
                           <div class="col-md-6">
                              <input type="text" class="form-control"
name="fatherName">
                           </div>
```

```
                    </div>

                    <div class="form-group row">
                      <label for="permanent_address" class="col-md-4 col-form-
label text-md-right">Mobile</label>
                      <div class="col-md-6">
                        <input type="text" class="form-control" name="mobile">
                      </div>
                    </div>
                      <div class="col-md-6 offset-md-4">
                        <button type="submit" class="btn btn-primary">
                        Save
                        </button>
                      </div>
                    </div>
                  </form>
                </div>
              </div>
          </div>
        </div>
    </main>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></scr
ipt>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script
>
</body>
  </div>

  <div id="Paris" class="w3-container w3-border city" style="display:none">
  <br>
  <link          href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
<script
src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <link rel="dns-prefetch" href="https://fonts.gstatic.com">
    <link          href="https://fonts.googleapis.com/css?family=Raleway:300,400,600"
rel="stylesheet" type="text/css">
```

```
   <link rel="icon" href="Favicon.png">
   <link                                                                rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">

</head>
<body>
<main class="my-form">
   <div class="cotainer">
      <div class="row justify-content-center">
         <div class="col-md-8">
               <div class="card">
                  <div class="card-header">Add Faculty</div>
                  <div class="card-body">
                     <form      name="my-form"      onsubmit="return      validform()"
action="addFaculty.jsp" method="post">
                        <div class="form-group row">
                           <label  for="full_name"  class="col-md-4  col-form-label  text-
md-right">Employee Id</label>
                           <div class="col-md-6">
                              <input type="text"class="form-control" name="empId">
                           </div>
                        </div>

                        <div class="form-group row">
                           <label  for="email_address"  class="col-md-4  col-form-label
text-md-right">Name</label>
                           <div class="col-md-6">
                              <input          type="text"                  class="form-control"
name="empName">
                           </div>
                        </div>

                        <div class="form-group row">
                           <label  for="user_name"  class="col-md-4  col-form-label  text-
md-right">Department</label>
                           <div class="col-md-6">
                              <input          type="text"                  class="form-control"
name="department">
                           </div>
                        </div>

                        <div class="form-group row">
                           <label  for="phone_number"  class="col-md-4  col-form-label
text-md-right">Mobile</label>
                           <div class="col-md-6">
                              <input type="text"  class="form-control" name="mobile">
                           </div>
                        </div>

                        <div class="form-group row">
```

```
                        <label for="present_address" class="col-md-4 col-form-label
text-md-right">Designation</label>
                        <div class="col-md-6">
                          <input type="text" class="form-control"
name="designation">
                        </div>
                      </div>


                        <div class="col-md-6 offset-md-4">
                          <button type="submit" class="btn btn-primary">
                           Save
                          </button>
                        </div>
                      </div>
                    </form>
                  </div>
                </div>
              </div>
            </div>
            <br>
          </div>

</main>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></scr
ipt>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script
>
</body>
</html>
 </div>

 <div id="Tokyo" class="w3-container w3-border city" style="display:none">

<section>
 <!--for demo wrap-->
 <div class="tbl-header">
  <table cellpadding="0" cellspacing="0" border="0">
   <thead>
    <tr>
     <th>Course Name</th>
      <th>Branch Name</th>
      <th>Roll Number</th>
      <th>Name</th>
      <th>Father Name</th>
      <th>Mobile</th>
```

```
      </tr>
     </thead>
   </table>
  </div>
  <div class="tbl-content">
   <table cellpadding="0" cellspacing="0" border="0">
    <tbody>
      <%@page import="project.ConnectionProvider" %>
<%@page import="java.sql.*"%>
<%
        try{
                Connection con = ConnectionProvider.getCon();
                Statement st = con.createStatement();
                ResultSet rst = st.executeQuery("select * from student");
                while(rst.next()){
                        %>

      <tr>
       <td><%=rst.getString(1) %></td>
       <td><%=rst.getString(2) %></td>
       <td><%=rst.getString(3) %></td>
       <td><%=rst.getString(4) %></td>
       <td><%=rst.getString(5) %></td>
       <td><%=rst.getString(6) %></td>
      </tr>

    </tbody>
    <%}
                }catch(Exception e){}
                %>
   </table>
  </div>
</section>
</div>
<div id="Tokyo1" class="w3-container w3-border city" style="display:none">

<section>
  <!--for demo wrap-->
  <div class="tbl-header">
   <table cellpadding="0" cellspacing="0" border="0">
    <thead>
      <tr>
       <th>Roll Number</th>
       <th>Date</th>
       <th>Time</th>


      </tr>
    </thead>
   </table>
```

```
  </div>
  <div class="tbl-content">
    <table cellpadding="0" cellspacing="0" border="0">
      <tbody>
      <%@page import="project.ConnectionProvider" %>
<%@page import="java.sql.*"%>
<%
          try{
                    Connection con = ConnectionProvider.getCon();
                    Statement st = con.createStatement();
                    ResultSet rst = st.executeQuery("select * from attendance");
                    while(rst.next()){
                              %>
       <tr>
          <td><%=rst.getString(1) %></td>
         <td><%=rst.getString(2) %></td>
         <td><%=rst.getString(3) %></td>

       </tr>
      </tbody>
      <%}}catch(Exception e){} %>
    </table>
  </div>
</section>


</body>
</html>
```

admin.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Admin Login</title>
  <link rel="stylesheet" type="text/css" href="assets/css/style.css">
  <style>
  .login-box{
    height: 380px;
}</style>
</head>
<body>
    <form method="post" action="adminLoginAction.jsp">
    <div class="login-box">
    <img src="./images/user.png" class="avatar">
       <h1>Admin Login</h1>
          <p>Username</p>
          <input type="text" name="username" placeholder="Enter Username"
required="required">
```

```
        <p>Password</p>
        <input type="password" name="password" placeholder="Enter Password"
required="required">
        <input type="submit" name="submit" value="Login">
        </form>
         <center><h1><a href="index.html">Back</a></h1></center>
      </div>
   </body>
</html>
```

adminLoginAction.jsp

```
<%
String userId = request.getParameter("username");
String pwd = request.getParameter("password");

if(userId.equalsIgnoreCase("admin") && pwd.equalsIgnoreCase("admin")){
        response.sendRedirect("adminHome.jsp");
} else {
        response.sendRedirect("errorAdminLogin.html");
}

%>
```

attendanceAction.jsp

```
<%@page import="project.ConnectionProvider" %>
<%@page import="com.blockchain.*" %>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@page import="java.sql.*,java.util.*"%>
<%
try{

        String rollNo = request.getParameter("rollNo");
        Connection con = ConnectionProvider.getCon();
        java.util.Date date=new java.util.Date();
        java.sql.Date sqlDate=new java.sql.Date(date.getTime());
        java.sql.Timestamp sqlTime=new java.sql.Timestamp(date.getTime());
        PreparedStatement ps=con.prepareStatement("insert into attendance
(rollNo,day,record) values(?,?,?)");
        ps.setString(1, rollNo);
        ps.setDate(2,sqlDate);
        ps.setTimestamp(3,sqlTime);
        ps.executeUpdate();
        System.out.println("Attendance recorded..");
```

```
        String res = rollNo + " : " + sqlDate + " , " + sqlTime;

        //Blockchain.addBlock(new Block("Genesis", "0"));

        Blockchain.addBlock(new
Block(res,Blockchain.chain.get(Blockchain.chain.size()-1).hash));
        System.out.println("\nThe blockchain: ");
        Blockchain.displayChain();

        response.sendRedirect("testHome.jsp");
}catch(Exception e){
        e.printStackTrace();
}
%>
```

errorAdminLogin.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <link rel="stylesheet" type="text/css" href="assets/css/style.css">
  <style>
  h3{
color:red;
}
  .login-box{
    height: 400px;
}</style>
</head>
<body>
   <form method="post" action="adminLoginAction.jsp">
   <div class="login-box">
   <img src="./images/user.png" class="avatar">
      <h3><center>Incorrect Username or Password</center></h3>
         <p>Username</p>
         <input type="text" name="username" placeholder="Enter Username"
required="required">
         <p>Password</p>
         <input type="password" name="password" placeholder="Enter Password"
required="required">
         <input type="submit" name="submit" value="Login">
         </form>
           <center><h1><a href="adminLogin.html">Back</a></h1></center>
      </div>
   </body>
</html>
```

errorAttendance.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <link rel="stylesheet" type="text/css" href="assets/css/style.css">
  <style>.login-box{
    height: 400px;

}</style>
</head>
<body></body>
    <form method="post" action="testHome.jsp">
    <div class="login-box">
    <center><img src="./images/attd.jpg" width="200" height="150"></center>
    <h3><center>Incorrect Roll Number <br>Try again</center></h3>

          <input type="text" name="rollNo" placeholder="Enter Roll Number"
required="required">
          <input type="submit" name="submit" value="Record Attendance">
          </form>

       </div>

</html>
```

errorFacultyLogin.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <link rel="stylesheet" type="text/css" href="assets/css/style.css">
  <style>
  h3{
color:red;
}
  .login-box{
    height: 400px;
}</style>
</head>
<body>
    <form method="post" action="facultyLoginAction.jsp">
    <div class="login-box">
    <img src="./images/teachers.png" class="avatar">
       <h3><center>Incorrect Username or Password</center></h3>
          <p>Username</p>
          <input type="text" name="username" placeholder="Enter Username"
required="required">
```

```
        <p>Password</p>
        <input type="password" name="password" placeholder="Enter Password" required="required">
        <input type="submit" name="submit" value="Login">
        </form>
          <center><h1><a href="facultyLogin.html">Back</a></h1></center>
      </div>
   </body>
</html>
```

errorReportView.html

```
<html>
<head>
<style>
h3{
color:red;
}
</style>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <link rel="stylesheet" type="text/css" href="assets/css/style.css">
</head>
<body></body>
   <form method="post" action="studentHome.jsp">
   <div class="login-box">
      <center><img src="./images/report.jpg" width="200" height="150"></center>
       <h3><center>Incorrect Roll Number <br>Try again</center></h3>

        <input type="text" name="rollNo" placeholder="Enter Roll Number" required="required">
        <input type="submit" name="submit" value="Search">
        </form>
          <center><h1><a href="index.html">Back</a></h1></center>
      </div>

</html>
```

errorStudentLogin.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <link rel="stylesheet" type="text/css" href="assets/css/style.css">
  <style>
  h3{
color:red;
}
  .login-box{
```

```
   height: 400px;
}</style>
</head>
<body>
   <form method="post" action="studentLoginAction.jsp">
   <div class="login-box">
   <img src="./images/stuUser.png" class="avatar">
      <h3><center>Incorrect Username or Password</center></h3>
         <p>Username</p>
         <input type="text" name="username" placeholder="Enter Username" required="required">
         <p>Password</p>
         <input type="password" name="password" placeholder="Enter Password" required="required">
         <input type="submit" name="submit" value="Login">
         </form>
          <center><h1><a href="StudentLogin.html">Back</a></h1></center>
      </div>
   </body>
</html>
```

facultyHome.jsp

```
<%@include file="header.html"%>
<!DOCTYPE html>
<html>
<title>Faculty Home</title>
<body>
<img src="./images/vishnuLogo.png"  align="left"width="100" height="100">

  <div class="w3-bar w3-black">
   <button class="w3-bar-item w3-button tablink w3-red" onclick="openCity(event,'London')">View StudentReport</button>
   <button class="w3-bar-item w3-button tablink" onclick="openCity(event,'Tokyo')">Student List</button>
    <button class="w3-bar-item w3-button tablink" onclick="openCity(event,'Tokyo1')">View Attendance</button>
    <a href="logoutAction.jsp" class="w3-bar-item w3-button tablink">Logout</a>
  </div>

  <div id="London" class="w3-container w3-border city">
  <br>
   <link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">
<script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<!------ Include the above in your HEAD tag ---------->
```

```
<!doctype html>
<html lang="en">
<head>
   <!-- Required meta tags -->
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

   <!-- Fonts -->
   <link rel="dns-prefetch" href="https://fonts.gstatic.com">
   <link href="https://fonts.googleapis.com/css?family=Raleway:300,400,600"
rel="stylesheet" type="text/css">



   <link rel="icon" href="Favicon.png">

   <!-- Bootstrap CSS -->
   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
</head>
<body>
        <jsp:include page="reportFacultyPage.html"/>


<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></scr
ipt>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script
>
</body>
</html>
  </div>

  <div id="Tokyo" class="w3-container w3-border city" style="display:none">

<section>
  <!--for demo wrap-->
  <div class="tbl-header">
    <table cellpadding="0" cellspacing="0" border="0">
      <thead>
        <tr>
         <th>Course Name</th>
          <th>Branch Name</th>
          <th>Roll Number</th>
          <th>Name</th>
          <th>Father Name</th>
```

```
        <th>Mobile</th>
       </tr>
     </thead>
   </table>
  </div>
  <div class="tbl-content">
   <table cellpadding="0" cellspacing="0" border="0">
     <tbody>
       <%@page import="project.ConnectionProvider" %>
<%@page import="java.sql.*"%>
<%
          try{
                    Connection con = ConnectionProvider.getCon();
                    Statement st = con.createStatement();
                    ResultSet rst = st.executeQuery("select * from student");
                    while(rst.next()){
                              %>

      <tr>
        <td><%=rst.getString(1) %></td>
        <td><%=rst.getString(2) %></td>
        <td><%=rst.getString(3) %></td>
        <td><%=rst.getString(4) %></td>
        <td><%=rst.getString(5) %></td>
        <td><%=rst.getString(6) %></td>
      </tr>

     </tbody>
     <%}
                    }catch(Exception e){}
                    %>
   </table>
  </div>
</section>
</div>


<div id="Tokyo1" class="w3-container w3-border city" style="display:none">

<section>
  <!--for demo wrap-->
  <div class="tbl-header">
   <table cellpadding="0" cellspacing="0" border="0">
     <thead>
       <tr>
         <th>Roll Number</th>
         <th>Date</th>
         <th>Time</th>
```

```
      </tr>
     </thead>
   </table>
  </div>
  <div class="tbl-content">
   <table cellpadding="0" cellspacing="0" border="0">
    <tbody>
    <%@page import="project.ConnectionProvider" %>
<%@page import="java.sql.*"%>
<%
         try{
                  Connection con = ConnectionProvider.getCon();
                  Statement st = con.createStatement();
                  ResultSet rst = st.executeQuery("select * from attendance");
                  while(rst.next()){
                        %>
      <tr>
        <td><%=rst.getString(1) %></td>
       <td><%=rst.getString(2) %></td>
       <td><%=rst.getString(3) %></td>

      </tr>
    </tbody>
    <%}}catch(Exception e){} %>
   </table>
  </div>
</section>


</body>
</html>
```

facultyLogin.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Faculty Login</title>
  <link rel="stylesheet" type="text/css" href="assets/css/style.css">
  <style>
  .login-box{
    height: 380px;
}</style>
</head>
<body>
   <form method="post" action="facultyLoginAction.jsp">
   <div class="login-box">
   <img src="./images/teachers.png" class="avatar">
      <h1>Faculty Login</h1>
```

```
        <p>Username</p>
        <input type="text" name="empId" placeholder="Enter Username" required="required">
        <p>Password</p>
        <input type="password" name="empName" placeholder="Enter Password" required="required">
        <input type="submit" name="s1" value="Login">
        </form>
         <center><h1><a href="index.html">Back</a></h1></center>

      </div>

   </body>
</html>
```

facultyLoginAction.jsp

```
<%@page import="project.ConnectionProvider" %>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@page import="java.sql.*,java.util.*"%>
<%
String empId=request.getParameter("empId");
session.setAttribute("empId",empId);
String empName=request.getParameter("empName");
session.setAttribute("empName", empName);
Connection con = ConnectionProvider.getCon();
Statement st= con.createStatement();
ResultSet rs=st.executeQuery("select * from faculty where empName='"+empName+"'
and empId='"+empId+"'");
try{
if(rs.next()){
if(rs.getString("empName").equals(empName)&&rs.getString("empId").equals(empI
d))
{
        response.sendRedirect("facultyHome.jsp");
}
}else{
        response.sendRedirect("errorFacultyLogin.html");
}
}
catch (Exception e) {
e.printStackTrace();
}
%>
```

facultyViewReport.jsp

```
<%@page import="project.ConnectionProvider" %>
```

```
<%@page import="java.sql.*" %>

<%
try{
        String rollNo = request.getParameter("rollNo");
        Connection con = ConnectionProvider.getCon();
        Statement st = con.createStatement();
        ResultSet rst = st.executeQuery("select * from student where
student.rollNo='"+rollNo+"'");
        if(rst.next())
        {
%>


<title><%=rst.getString(4)%> Report</title>
<body>
<img src="./images/vishnuLogo.png"  align="left"width="100" height="100">



  <hr class="new1">
<style>
table{
  width:100%;
  table-layout: fixed;
}
th{
  padding: 20px 15px;
  text-align: left;
  font-weight: 500;
  font-size: 12px;
  color: #fff;
  text-transform: uppercase;
 border: 2px solid rgba(255,255,255,0.3);
}


/* demo styles */

@import url(https://fonts.googleapis.com/css?family=Roboto:400,500,300,700);
body{
  background: -webkit-linear-gradient(left, #c938fa, #2dd2d2);
  background: linear-gradient(to right, #c938fa, #2dd2d2);
  font-family: 'Roboto', sans-serif;
}

</style>
  <!--for demo wrap-->
  <div class="tbl-header">
    <table cellpadding="0" cellspacing="0" border="0">
```

```
    <thead>
      <tr>
        <th>institution Name: BVRITH</th>
        <th>Course Name: <%=rst.getString(1) %></th>
        <th>Branch Name: <%=rst.getString(2) %></th>
        <th><center>RollNo: <%=rst.getString(3) %></center></th>
      </tr>
    </thead>
    <thead>
      <tr>
        <th>Name: <%=rst.getString(4) %></th>
        <th>Father Name:<%=rst.getString(5) %></th>
        <th>Mobile:<%=rst.getString(6) %></th>
        <th><a titlt="print screen" alt="print screen" onclick="window.print();"
target="_blank" style="cursor:pointer;"><center><img
src="./images/print.png"></center></a></th>
      </tr>
    </thead>
   </table>
  </div>
<style>
html {
  font-family:arial;
  font-size: 25px;
}

td {
  border: 2px solid #726E6D;
  padding: 15px;
  color:black;
  text-align:center;
}

thead{
  font-weight:bold;
  text-align:center;
  background: #625D5D;
  color:white;
}

table {
  border-collapse: collapse;
}

.footer {
  text-align:right;
  font-weight:bold;
}

tbody >tr:nth-child(odd) {
```

```
 background: #D1D0CE;
}

</style>
<head>
 <hr class="new1">
</head>
<body>
 <table>
 <%
        } else{
                response.sendRedirect("errorDgiOneView.html");
        }
        ResultSet rst1 = st.executeQuery("SELECT COUNT(*), rollNo FROM
attendance where attendance.rollNo='"+rollNo+"'");
        if(rst1.next()){


 %>
  <thead>
   <tr>

    <td rowspan="2">Total Held</td>
    <td rowspan="2">Classes Attended</td>
    <td rowspan="2">Percentage</td>

   </tr>

  </thead>
  <tbody>
   <tr>


    <td>35</td>
    <td><%=rst1.getString(1) %></td>
    <td><%int res = Integer.parseInt(rst1.getString(1)); out.println((res*100)/10);
%></td>
   </tr>
  </tbody>

 </table>

 <br/>
 <br/>

  <a href="facultyHome.jsp">Back</a>
</body>

<%
        }else{
```

```
                response.sendRedirect("errorDgiOneView.html");
        }
        } catch(Exception e){}

%>
```

header.html

```
<script>
function openCity(evt, cityName) {
  var i, x, tablinks;
  x = document.getElementsByClassName("city");
  for (i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
  tablinks = document.getElementsByClassName("tablink");
  for (i = 0; i < x.length; i++) {
    tablinks[i].className = tablinks[i].className.replace(" w3-red", "");
  }
  document.getElementById(cityName).style.display = "block";
  evt.currentTarget.className += " w3-red";
}
</script>

 <style>

h1{
  font-size: 30px;
  color: #fff;
  text-transform: uppercase;
  font-weight: 300;
  text-align: center;
  margin-bottom: 15px;
}
table{
  width:100%;
  table-layout: fixed;
}
.tbl-header{
  background-color: rgba(255,255,255,0.3);
 }
.tbl-content{
  height:300px;
  overflow-x:auto;
  margin-top: 0px;
  border: 1px solid rgba(255,255,255,0.3);
}
th{
  padding: 20px 15px;
```

```
  text-align: left;
  font-weight: 500;
  font-size: 12px;
  color: #fff;
  text-transform: uppercase;
}
td{
  padding: 15px;
  text-align: left;
  vertical-align:middle;
  font-weight: 300;
  font-size: 12px;
  color: #fff;
  border-bottom: solid 1px rgba(255,255,255,0.1);
}

@import url(https://fonts.googleapis.com/css?family=Roboto:400,500,300,700);
body{
   background: -webkit-linear-gradient(left, #c938fa, #2dd2d2);
  background: linear-gradient(to right, #c938fa, #2dd2d2);
  font-family: 'Roboto', sans-serif;
}
section{
  margin: 50px;
}

.made-with-love {
  margin-top: 40px;
  padding: 10px;
  clear: left;
  text-align: center;
  font-size: 10px;
  font-family: arial;
  color: #fff;
}
.made-with-love i {
  font-style: normal;
  color: #F50057;
  font-size: 14px;
  position: relative;
  top: 2px;
}
.made-with-love a {
  color: #fff;
  text-decoration: none;
}
.made-with-love a:hover {
  text-decoration: underline;
}
```

```
::-webkit-scrollbar {
    width: 6px;
}
::-webkit-scrollbar-track {
    -webkit-box-shadow: inset 0 0 6px rgba(0,0,0,0.3);
}
::-webkit-scrollbar-thumb {
    -webkit-box-shadow: inset 0 0 6px rgba(0,0,0,0.3);
  }</style>

<style>
html{
  box-sizing:border-box
  }
*,*:before,*:after{
  box-sizing:inherit
}
html{
  -ms-text-size-adjust:100%;
  -webkit-text-size-adjust:100%
  }
body{
  margin:0
}
article,aside,details,figcaption,figure,footer,header,main,menu,nav,section{
  display:block
  }
summary{
    display:list-item
}
audio,canvas,progress,video{
  display:inline-block
  }
progress{
  vertical-align:baseline
}
audio:not([controls]){
  display:none;height:0
}
[hidden],template{
  display:none
}
a{
  background-color:transparent
}
a:active,a:hover{
  outline-width:0
}
abbr[title]{
```

```
  border-bottom:none;
  text-decoration:underline;
  text-decoration:underline dotted
}
b,strong{
  font-weight:bolder
}
dfn{
  font-style:italic
}mark{
  background:#ff0;
  color:#000
}
small{
  font-size:80%
}
```

logoutAction.jsp

```
<% session.removeAttribute("name");
session.removeAttribute("rollNo");
session.removeAttribute("empId");
session.removeAttribute("empName");
session.invalidate();

response.sendRedirect("index.html");
%>
```

RecordAttendance.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <link rel="stylesheet" type="text/css" href="assets/css/style.css">
  <style>.login-box{
    height: 400px;

}</style>
</head>
<body></body>
    <form method="post" action="attendanceAction.jsp">
    <div class="login-box">
    <center><img src="./images/attd.jpg" width="200" height="150"></center>

          <input type="text" name="rollNo" placeholder="Enter Roll Number"
required="required">
          <input type="submit" name="submit" value="Record Attendance">
          </form>
```

```
    </div>

</html>
```

report.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <link rel="stylesheet" type="text/css" href="assets/css/style.css">
  <style>.login-box{
    height: 400px;

}</style>
</head>
<body></body>
    <form method="post" action="studentHome.jsp">
    <div class="login-box">
    <center><img src="./images/report.jpg" width="200" height="150"></center>

          <input type="text" name="rollNo" placeholder="Enter Roll Number"
required="required">
          <input type="submit" name="submit" value="Get Report">
          </form>

      </div>

</html>
```

reportFacultyPage.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <link rel="stylesheet" type="text/css" href="assets/css/style.css">
  <style>.login-box{
    height: 400px;

}</style>
</head>
<body></body>
    <form method="post" action="facultyViewReport.jsp">
    <div class="login-box">
    <center><img src="./images/report.jpg" width="200" height="150"></center>

          <input type="text" name="rollNo" placeholder="Enter Roll Number"
required="required">
          <input type="submit" name="submit" value="Get Report">
```

```
        </form>

      </div>

</html>
```

studentHome.jsp

```
<%@page import="project.ConnectionProvider" %>
<%@page import="java.sql.*" %>

<%
try{
        String user = request.getParameter("rollNo");
        String rollNo = (String)session.getAttribute("rollNo");
        Connection con = ConnectionProvider.getCon();
        Statement st = con.createStatement();
        ResultSet rst = st.executeQuery("select * from student inner join attendance
where student.rollNo=attendance.rollNo and student.rollNo='"+rollNo+"'");
        if(rst.next()) {
                if(user.equalsIgnoreCase(rollNo))
                {
%>


<title><%=rst.getString(4)%> Report</title>
<body>
<img src="./images/vishnuLogo.png"  align="left"width="100" height="100">



  <hr class="new1">
<style>
table{
  width:100%;
  table-layout: fixed;
}
th{
  padding: 20px 15px;
  text-align: left;
  font-weight: 500;
  font-size: 12px;
  color: #fff;
  text-transform: uppercase;
 border: 2px solid rgba(255,255,255,0.3);
}


/* demo styles */
```

```
@import url(https://fonts.googleapis.com/css?family=Roboto:400,500,300,700);
body{
  background: -webkit-linear-gradient(left, #c938fa, #2dd2d2);
  background: linear-gradient(to right, #c938fa, #2dd2d2);
  font-family: 'Roboto', sans-serif;
}

</style>
  <!--for demo wrap-->
  <div class="tbl-header">
    <table cellpadding="0" cellspacing="0" border="0">
      <thead>
        <tr>
          <th>institution Name: BVRITH</th>
          <th>Course Name: <%=rst.getString(1) %></th>
          <th>Branch Name: <%=rst.getString(2) %></th>
          <th><center>RollNo: <%=rst.getString(3) %></center></th>
        </tr>
      </thead>
      <thead>
        <tr>
          <th>Name: <%=rst.getString(4) %></th>
          <th>Father Name:<%=rst.getString(5) %></th>
          <th>Mobile:<%=rst.getString(6) %></th>
          <th><a titlt="print screen" alt="print screen" onclick="window.print();"
target="_blank" style="cursor:pointer;"><center><img
src="./images/print.png"></center></a></th>
        </tr>
      </thead>
    </table>
  </div>
<style>
html {
  font-family:arial;
  font-size: 25px;
}

td {
  border: 2px solid #726E6D;
  padding: 15px;
  color:black;
  text-align:center;
}

thead{
  font-weight:bold;
  text-align:center;
  background: #625D5D;
  color:white;
```

```
}

table {
  border-collapse: collapse;
}

.footer {
  text-align:right;
  font-weight:bold;
}

tbody >tr:nth-child(odd) {
  background: #D1D0CE;
}

</style>
<head>
  <hr class="new1">
</head>
<body>
  <table>
  <%

          } else{
                  response.sendRedirect("errorReportView.html");
          }
          }
          ResultSet rst1 = st.executeQuery("SELECT COUNT(*), rollNo FROM
attendance where attendance.rollNo='"+rollNo+"'");
          if(rst1.next()){


 %>
   <thead>
     <tr>

       <td rowspan="2">Total Held</td>
       <td rowspan="2">Classes Attended</td>
       <td rowspan="2">Percentage</td>

     </tr>

   </thead>
   <tbody>
     <tr>


       <td>35</td>
       <td><%=rst1.getString(1) %></td>
```

```
      <td><%int res = Integer.parseInt(rst1.getString(1)); out.println((res*100)/10);
%></td>
     </tr>
   </tbody>

  </table>

  <br/>
  <br/>

   <a href="testHome.jsp">Back</a>
</body>

<%
          }else{
                    response.sendRedirect("errorReportView.html");
          }
          } catch(Exception e){}

%>
```

StudentLogin.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Student Login</title>
  <link rel="stylesheet" type="text/css" href="assets/css/style.css">
  <style>
  .login-box{
    height: 380px;
}</style>
</head>
<body>
   <form method="post" action="studentLoginAction.jsp">
   <div class="login-box">
   <img src="./images/stuUser.png" class="avatar">
      <h1>Student Login</h1>
         <p>Username</p>
         <input type="text" name="name" placeholder="Enter Username"
required="required">
         <p>Password</p>
         <input type="password" name="rollNo" placeholder="Enter Password"
required="required">
         <input type="submit" name="s1" value="Login">
         </form>
          <center><h1><a href="index.html">Back</a></h1></center>

      </div>
```

```
   </body>
</html>
```

studentLoginAction.jsp

```
<%@page import="project.ConnectionProvider" %>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@page import="java.sql.*,java.util.*"%>
<%
String name=request.getParameter("name");
session.setAttribute("name",name);
String rollNo=request.getParameter("rollNo");
session.setAttribute("rollNo", rollNo);
Connection con = ConnectionProvider.getCon();
Statement st= con.createStatement();
ResultSet rs=st.executeQuery("select * from student where name='"+name+"' and
rollNo='"+rollNo+"'");
try{
if(rs.next()){
if(rs.getString("rollNo").equals(rollNo)&&rs.getString("name").equals(name))
{
        response.sendRedirect("testHome.jsp");
}
}else{
        response.sendRedirect("errorStudentLogin.html");
}
}
catch (Exception e) {
e.printStackTrace();
}
%>
```

testHome.jsp

```
<%@include file="header.html"%>
<!DOCTYPE html>
<html>
<title>Student Home</title>
<body>
<img src="./images/vishnuLogo.png"  align="left"width="100" height="100">

  <div class="w3-bar w3-black">
   <button class="w3-bar-item w3-button tablink w3-red"
onclick="openCity(event,'London')">Report</button>
   <button class="w3-bar-item w3-button tablink"
onclick="openCity(event,'Paris')">Attendance</button>
```

```
    <button class="w3-bar-item w3-button tablink" onclick="openCity(event,'Tokyo1')">View Attendance</button>
    <a href="logoutAction.jsp" class="w3-bar-item w3-button tablink">Logout</a>
  </div>

  <div id="London" class="w3-container w3-border city">
  <br>
   <link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">
<script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<!------ Include the above in your HEAD tag ---------->

<!doctype html>
<html lang="en">
<head>
   <!-- Required meta tags -->
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

   <!-- Fonts -->
   <link rel="dns-prefetch" href="https://fonts.gstatic.com">
   <link href="https://fonts.googleapis.com/css?family=Raleway:300,400,600" rel="stylesheet" type="text/css">



   <link rel="icon" href="Favicon.png">

   <!-- Bootstrap CSS -->
   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
</head>
<body>
        <jsp:include page="report.html"/>




<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
```

```
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script
>
</body>
  </div>

  <div id="Paris" class="w3-container w3-border city" style="display:none">
  <br>
  <link          href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
<script
src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<!------ Include the above in your HEAD tag ---------->

<!doctype html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Fonts -->
    <link rel="dns-prefetch" href="https://fonts.gstatic.com">
    <link          href="https://fonts.googleapis.com/css?family=Raleway:300,400,600"
rel="stylesheet" type="text/css">



    <link rel="icon" href="Favicon.png">

    <!-- Bootstrap CSS -->
    <link                                                                rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">

</head>
<body>
<jsp:include page="RecordAttendance.html"/>



<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></scr
ipt>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script
>
</body>
```

```
</html>
  </div>

  <div id="Tokyo" class="w3-container w3-border city" style="display:none">

<section>
  <!--for demo wrap-->
  <div class="tbl-header">
    <table cellpadding="0" cellspacing="0" border="0">
      <thead>
        <tr>
         <th>Course Name</th>
          <th>Branch Name</th>
          <th>Roll Number</th>
          <th>Name</th>
          <th>Father Name</th>
          <th>Mobile</th>
        </tr>
      </thead>
    </table>
  </div>
  <div class="tbl-content">
    <table cellpadding="0" cellspacing="0" border="0">
      <tbody>
        <%@page import="project.ConnectionProvider" %>
<%@page import="java.sql.*"%>
<%
          try{
                    Connection con = ConnectionProvider.getCon();
                    Statement st = con.createStatement();
                    ResultSet rst = st.executeQuery("select * from student");
                    while(rst.next()){
                              %>

       <tr>
         <td><%=rst.getString(1) %></td>
         <td><%=rst.getString(2) %></td>
         <td><%=rst.getString(3) %></td>
         <td><%=rst.getString(4) %></td>
         <td><%=rst.getString(5) %></td>
         <td><%=rst.getString(6) %></td>
       </tr>

     </tbody>
     <%}
                    }catch(Exception e){}
                    %>
    </table>
  </div>
</section>
```

```
</div>
<div id="Tokyo1" class="w3-container w3-border city" style="display:none">

<section>
  <!--for demo wrap-->
  <div class="tbl-header">
    <table cellpadding="0" cellspacing="0" border="0">
      <thead>
        <tr>
          <th>Roll Number</th>
          <th>Date</th>
          <th>Time</th>


        </tr>
      </thead>
    </table>
  </div>
  <div class="tbl-content">
    <table cellpadding="0" cellspacing="0" border="0">
      <tbody>
      <%@page import="project.ConnectionProvider" %>
<%@page import="java.sql.*"%>
<%
          try{
                    String rollNo=(String)session.getAttribute("rollNo");
                    Connection con = ConnectionProvider.getCon();
                    Statement st = con.createStatement();
                    ResultSet rst = st.executeQuery("select * from attendance where
rollNo='"+rollNo+"'");
                    while(rst.next()){
                              %>
        <tr>
          <td><%=rst.getString(1) %></td>
          <td><%=rst.getString(2) %></td>
          <td><%=rst.getString(3) %></td>

        </tr>
      </tbody>
      <%}}catch(Exception e){} %>
    </table>
  </div>
</section>


</body>
</html>
```

Style.css

```
@import url(https://fonts.googleapis.com/css?family=Roboto:400,500,300,700);
body{
  background: -webkit-linear-gradient(left, #c938fa, #2dd2d2);
  background: linear-gradient(to right, #c938fa, #2dd2d2);
  font-family: 'Roboto', sans-serif;
}body{
    margin: 0;
    padding: 0;
    background-size: cover;
    background-position: center;
    font-family: sans-serif;
}


.login-box{
    width: 320px;
    height: 450px;
    background: white;
    color: #fff;
    top: 50%;
    left: 50%;
    position: absolute;
    transform: translate(-50%,-50%);
   box-sizing: border-box;
    padding: 70px 30px;
}

.avatar{
    width: 100px;
    height: 100px;
    position: absolute;
    top: -50px;
    left: calc(50% - 50px);
}
h1{
    margin: 0;
    padding: 0 0 20px;
    text-align: center;
    font-size: 22px;
    color:black;
}
.login-box p{
    margin: 0;
    padding: 0;
    font-weight: bold;
}
```

```
.login-box input{
   width: 100%;
   margin-bottom: 20px;
}
.login-box                                                    input[type="text"],
input[type="password"],input[type="number"],input[type="email"]
{
   border: none;
   border-bottom: 1px solid black;
   background: transparent;
   outline: none;
   height: 40px;
   color: black;
   font-size: 16px;
}
.login-box input[type="submit"]
{
   border: none;
   outline: none;
   height: 40px;
   background: #1c8adb;
   color: black;
   font-size: 18px;
   border-radius: 20px;
}
.login-box input[type="submit"]:hover
{
   cursor: pointer;
   background: #39dc79;
   color: #000;
}

.login-box a{
   text-decoration: none;
   font-size: 14px;
   color: black;
}
.login-box a:hover
{
   color: #39dc79;
}
h2
{
        color:black;
}
```

main.css

```
@charset "UTF-8";
@import url(font-awesome.min.css);
@import "https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,600";


	html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p, blockquote, pre, a, abbr, acronym, address, big, cite, code, del, dfn, em, img, ins, kbd, q, s, samp, small, strike, strong, sub, sup, tt, var, b, u, i, center, dl, dt, dd, ol, ul, li, fieldset, form, label, legend, table, caption, tbody, tfoot, thead, tr, th, td, article, aside, canvas, details, embed, figure, figcaption, footer, header, hgroup, menu, nav, output, ruby, section, summary, time, mark, audio, video {
		margin: 0;
		padding: 0;
		border: 0;
		font-size: 100%;
		font: inherit;
		vertical-align: baseline;
	}


	article, aside, details, figcaption, figure, footer, header, hgroup, menu, nav, section {
		display: block;
	}
	#nav {
		color:white;
	}

	body {
		line-height: 1;
	}

	ol, ul {
		list-style: none;
	}

	blockquote, q {
		quotes: none;
	}

	blockquote:before, blockquote:after, q:before, q:after {
		content: '';
		content: none;
	}

	table {
		border-collapse: collapse;
```

```
        border-spacing: 0;
}

body {
        -webkit-text-size-adjust: none;
}


*, *:before, *:after {
        -moz-box-sizing: border-box;
        -webkit-box-sizing: border-box;
        box-sizing: border-box;
}

.container {
        margin-left: auto;
        margin-right: auto;
}

.container.\31 25\25 {
        width: 100%;
        max-width: 100em;
        min-width: 80em;
}

.container.\37 5\25 {
        width: 60em;
}

.container.\35 0\25 {
        width: 40em;
}

.container.\32 5\25 {
        width: 20em;
}

.container {
        width: 80em;
}

@media screen and (max-width: 1680px) {

        .container.\31 25\25 {
                width: 100%;
                max-width: 100em;
                min-width: 80em;
        }

        .container.\37 5\25 {
```

```
                width: 60em;
        }

        .container.\35 0\25 {
                width: 40em;
        }

        .container.\32 5\25 {
                width: 20em;
        }

        .container {
                width: 80em;
        }

}

@media screen and (max-width: 1280px) {

        .container.\31 25\25 {
                width: 100%;
                max-width: 81.25em;
                min-width: 65em;
        }

        .container.\37 5\25 {
                width: 48.75em;
        }

        .container.\35 0\25 {
                width: 32.5em;
        }

        .container.\32 5\25 {
                width: 16.25em;
        }

        .container {
                width: 65em;
        }

}

@media screen and (max-width: 980px) {

        .container.\31 25\25 {
                width: 100%;
                max-width: 112.5%;
                min-width: 90%;
        }
```

```
		.container.\37 5\25 {
			width: 67.5%;
		}

		.container.\35 0\25 {
			width: 45%;
		}

		.container.\32 5\25 {
			width: 22.5%;
		}

		.container {
			width: 90%;
		}

	}
}
```

main.js

```
(function($) {

		skel.breakpoints({
			xlarge: '(max-width: 1680px)',
			large:   '(max-width: 1280px)',
			medium:         '(max-width: 980px)',
			small:  '(max-width: 736px)',
			xsmall:'(max-width: 480px)'
		});

	$(function() {

		var	$window = $(window),
			$body = $('body');
			$body.addClass('is-loading');

			$window.on('load', function() {
				window.setTimeout(function() {
					$body.removeClass('is-loading');
				}, 100);
			});

			skel.on('+medium -medium', function() {
				$.prioritize(
					'.important\\28 medium\\29',
					skel.breakpoint('medium').active
				);
```

```
                        });

                        $(
                                '<div id="navPanel">' +
                                        $('#nav').html() +
                                        '<a href="#navPanel" class="close"></a>' +
                                '</div>'
                        )
                                .appendTo($body)
                                .panel({
                                        delay: 500,
                                        hideOnClick: true,
                                        hideOnSwipe: true,
                                        resetScroll: true,
                                        resetForms: true,
                                        side: 'left'
                                });
                        if (skel.vars.os == 'wp' && skel.vars.osVersion < 10)
                                $('#navPanel')
                                        .css('transition', 'none');

        });

})(jQuery);
```

Block.java

```
package com.blockchain;

import java.util.Date;

public class Block {

        public String hash;
        public String previousHash;
        private String data;
        private long timeStamp;
        private int nonce;

        //Block Constructor.
        public Block(String data,String previousHash ) {
                this.data = data;
                this.previousHash = previousHash;
                this.timeStamp = new Date().getTime();

                this.hash = calculateHash();
        }

        public String getHash() {
```

```
		return hash;
	}

	public void setHash(String hash) {
		this.hash = hash;
	}

	public String getPreviousHash() {
		return previousHash;
	}

	public void setPreviousHash(String previousHash) {
		this.previousHash = previousHash;
	}

	public String getData() {
		return data;
	}

	public void setData(String data) {
		this.data = data;
	}

	public long getTimeStamp() {
		return timeStamp;
	}

	public void setTimeStamp(long timeStamp) {
		this.timeStamp = timeStamp;
	}

	public int getNonce() {
		return nonce;
	}

	public void setNonce(int nonce) {
		this.nonce = nonce;
	}

	public String calculateHash() {
		String calculatedhash = Hashing.applySha256(
					previousHash +
					Long.toString(timeStamp) +
					Integer.toString(nonce) +
					data
					);
		return calculatedhash;
	}

	//Increases nonce value until hash target is reached.
```

```
	public void mineBlock(int difficulty) {
		String target = Hashing.getDificultyString(difficulty);
		while(!hash.substring( 0, difficulty).equals(target)) {
			nonce ++;
			hash = calculateHash();
		}
		System.out.println("Block Mined with nounce "+ nonce+" and " +
hash);
	}

}
```

Blockchain.java

```
package com.blockchain;
import java.util.ArrayList;


import com.google.gson.GsonBuilder;

public class Blockchain {

	public static ArrayList<Block> chain = new ArrayList<Block>();

	public static int difficulty = 5;



	public static Boolean isChainValid() {
		Block currentBlock;
		Block previousBlock;
		String hashTarget = new String(new char[difficulty]).replace('\0', '0');

		//loop through blockchain to check hashes:
		for(int i=1; i < chain.size(); i++) {
			currentBlock = chain.get(i);
			previousBlock = chain.get(i-1);
			//compare registered hash and calculated hash:
			if(!currentBlock.hash.equals(currentBlock.calculateHash()) ){
				System.out.println("Current Hashes not equal");

				return false;
			}
			//compare previous hash and registered previous hash
			if(!previousBlock.hash.equals(currentBlock.previousHash) ) {
				System.out.println("Previous Hashes not equal");
				return false;
			}
			//check if hash is solved
```

```
                        if(!currentBlock.hash.substring( 0, difficulty).equals(hashTarget)) {
                                System.out.println("This block hasn't been mined");
                                return false;
                        }

                }
                return true;
        }

        public static void addBlock(Block newBlock) {
                newBlock.mineBlock(difficulty);
                chain.add(newBlock);
        }

public static void displayChain() {

                for(int i=0; i<chain.size(); i++) {
                        System.out.println("Block: " + i);
                        System.out.println("Data: " + chain.get(i).getData());
                        System.out.println("Timestamp: " + chain.get(i).getTimeStamp());
                        System.out.println("PreviousHash: " + chain.get(i).getPreviousHash());
                        System.out.println("Hash: " + chain.get(i).getHash());
                        System.out.println("Nonce: " + chain.get(i).getNonce());
                        System.out.println();
                }

        }
}
```

Hashing.java

```
package com.blockchain;
import java.security.MessageDigest;

import com.google.gson.GsonBuilder;

public class Hashing {

        //Applies Sha256 to a string and returns the result.
        public static String applySha256(String input){

                try {
                        MessageDigest digest = MessageDigest.getInstance("SHA-256");

                        //Applies sha256 to our input,
```

```
                byte[] hash = digest.digest(input.getBytes("UTF-8"));

                StringBuffer hexString = new StringBuffer(); // This will contain
hash as hexidecimal
                for (int i = 0; i < hash.length; i++) {
                        String hex = Integer.toHexString(0xff & hash[i]);
                        if(hex.length() == 1) hexString.append('0');
                        hexString.append(hex);
                }
                return hexString.toString();
            }
            catch(Exception e) {
                throw new RuntimeException(e);
            }
        }

        //Short hand helper to turn Object into a json string
        public static String getJson(Object o) {
            return new GsonBuilder().setPrettyPrinting().create().toJson(o);
        }

        //Returns difficulty string target, to compare to hash. eg difficulty of 5 will
return "00000"
        public static String getDificultyString(int difficulty) {
            return new String(new char[difficulty]).replace('\0', '0');
        }


}
```

ConnectionProvider.java

```
package project;
import java.sql.Connection;
import java.sql.DriverManager;

public class ConnectionProvider {
        public static Connection getCon() {
            try {
                Class.forName("com.mysql.jdbc.Driver");
                Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/project1", "root",
"manasa12");
                return con;
            }
            catch(Exception e) {
                System.out.println(e);
                return null;
            }
```

```
        }
}
```

4.2 Testing

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of software as a system element and attendant costs associated with a software failure are motivating factors for, we planned, through testing. Testing is the process of executing a program with the intent of finding an error. The design of tests for software and other engineered products can be as challenging as the initial design of the product itself.

There of basically two types of testing approaches.

One is Black-Box testing – the specified function that a product has been designed to perform, tests can be conducted that demonstrate each function is fully operated. The other is White-Box testing – knowing the internal workings of the product ,tests can be conducted to ensure that the internal operation of the product performs according to specifications and all internal components have been adequately exercised.

White box and Black box testing methods have been used to test this package. The entire loop constructs have been tested for their boundary and intermediate conditions. The test data was designed with a view to check for all the conditions and logical decisions. Error handling has been taken care of by the use of exception handlers.

4.2.1 Testing Strategies

Testing is a set of activities that can be planned in advanced and conducted systematically. A strategy for software testing must accommodation low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

Software testing is one element of verification and validation. Verification refers to the set of activities that ensure that software correctly implements as specific function. Validation refers to a different set of activities that ensure that the software that has been built is traceable to customer requirements.

The main objective of software is testing to uncover errors. To fulfill this objective, a series of test steps unit, integration, validation and system tests are planned and executed. Each test step is accomplished through a series of systematic test technique that assist in the design of test cases. With each testing step, the level of abstraction with which software is considered is broadened.

Testing is the only way to assure the quality of software and it is an umbrella activity rather than a separate phase. This is an activity to be performed in parallel with the software effort and one that consists of its own phases of analysis, design, implementation, execution and maintenance.

UNIT TESTING:

This testing method considers a module as single unit and checks the unit at interfaces and communicates with other modules rather than getting into details at statement level. Here the module will be treated as a black box, which will take some input and generate output. Outputs for a given set of input combination are pre-calculated and are generated by the module.

SYSTEM TESTING:

Here all the pre tested individual modules will be assembled to create the larger system and tests are carried out at system level to make sure that all modules are working in synchronous with each other. This testing methodology helps in making sure that all modules which are running perfectly when checked individually are also running in cohesion with other modules. For this testing we create test cases to check all modules once and then generated test combinations of test paths throughout the system to make sure that no path is making its way into chaos.

INTEGRATED TESTING

Testing is a major quality control measure employed during software development. Its basic function is to detect errors. Sub functions when combined may not produce than it is desired. Global data structures can represent the problems. Integrated testing is a systematic technique for constructing the program structure while conducting the tests. To uncover errors that are associated with interfacing the objective is to make unit test modules and built a program structure that has been detected by design. In a non -

incremental integration all the modules are combined in advance and the program is tested as a whole. Here errors will appear in an endless loop function. In incremental testing the program is constructed and tested in small segments where the errors are isolated and corrected.

Different incremental integration strategies are top – down integration, bottom – up integration, regression testing.

REGRESSION TESTING

Each time a new module is added as a part of integration as the software changes. Regression testing is an actually that helps to ensure changes that do not introduce unintended behavior as additional errors.

Regression testing maybe conducted manually by executing a subset of all test cases or using automated capture play back tools enable the software engineer to capture the test case and results for subsequent playback and compression. The regression suit contains different classes of test cases.

A representative sample to tests that will exercise all software functions.

Additional tests that focus on software functions that are likely to be affected by the change.

4.3 TEST CASES

Integrated and regression testing strategies are used in this application for testing.

Test Case Id	Test Scenario	Expected Result	Actual result	Pass/Fail
TC01	Check whether application is working fine after copying url	Index Page should be displayed	As Expected	Pass
TC02	Check whether navbar is working fine	Respective login pages open	As Expected	Pass
TC03	Check whether login is done	Navigated to respective Home pages	As Expected	Pass
TC04	Check whether registration of student is done	Data stored in database	As Expected	Pass
TC05	Check whether registration of faculty is done	Data stored in database	As Expected	Pass
TC06	Check whether attendance is stored in blockchain	Data stored in Blockchain and displayed	As Expected	Pass
TC07	Check whether student can view their attendance	Logged in Student's time records is displayed	As Expected	Pass
TC08	Check whether student can view report	Logged in Student's report is displayed	As Expected	Pass
TC09	Check whether student is able to download report	Print dialog box with report is opened	As Expected	Pass
TC10	Check whether admin can view student list	All registered students details is displayed	As Expected	Pass
TC11	Check whether admin can view all students attendance	All registered students time records are displayed	As Expected	Pass
TC12	Check whether faculty can view student report	Only the given student roll number report is displayed	As Expected	Pass
TC13	Check whether faculty is able to download student report	Print dialog box with report is opened	As Expected	Pass
TC14	Check whether faculty can view all students attendance	All registered students time records are displayed	As Expected	Pass
TC15	Check whether logout is working	Session ends and navigated to Index page	As Expected	Pass

Fig 4.3: Test cases

4.4 APPLICATION RUNNING SCREENSHOTS

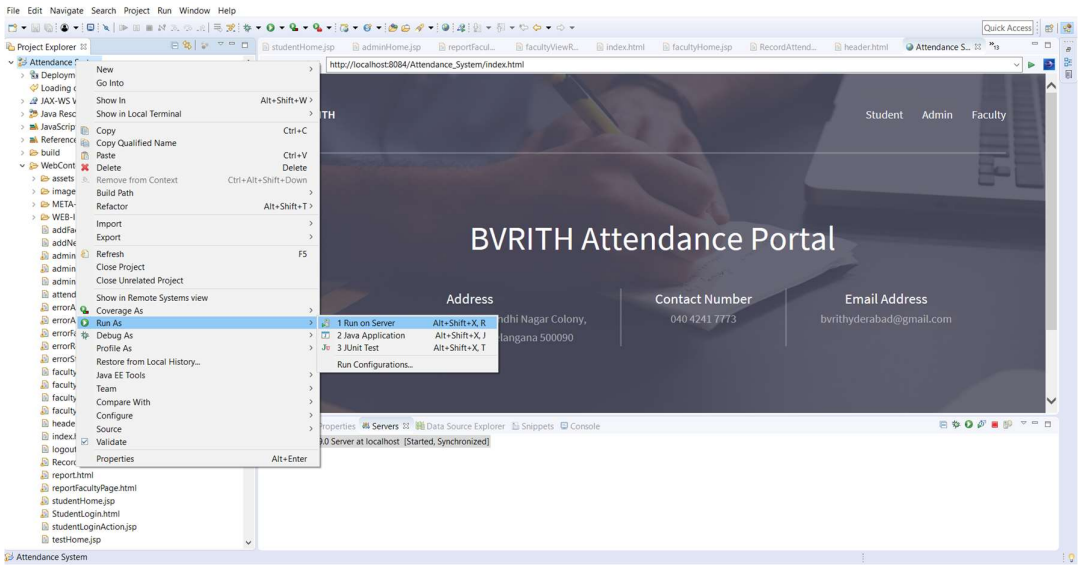

Fig 4.4.1: Running the project

4.5 DATABASE TABLES SCREENSHOTS

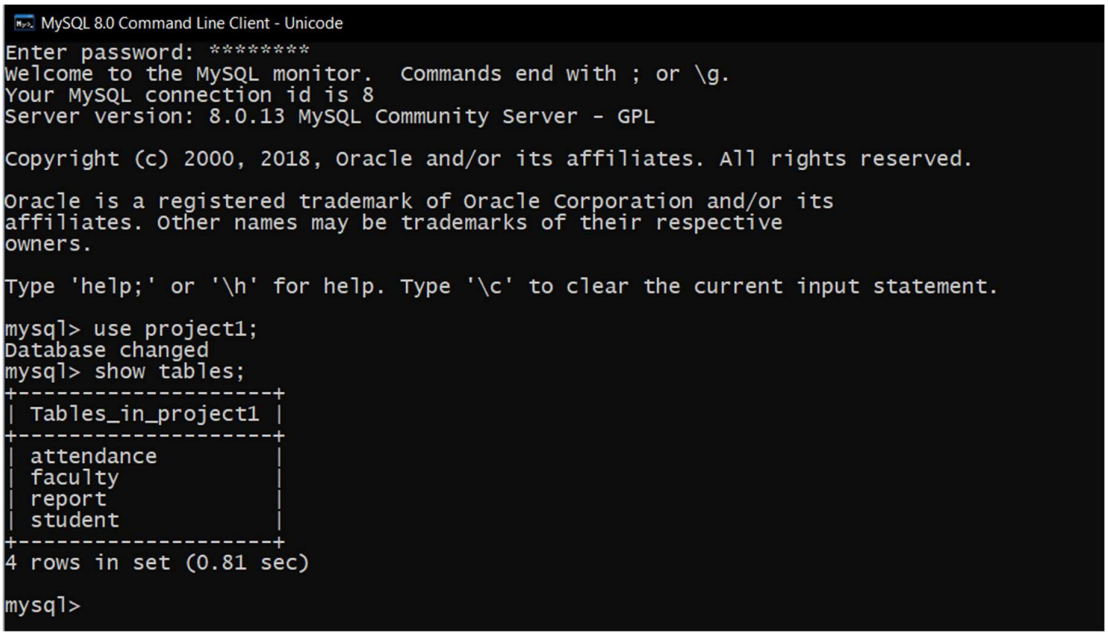

Fig 4.5.1: Tables in the project

4.6 OUTPUT SCREENSHOTS

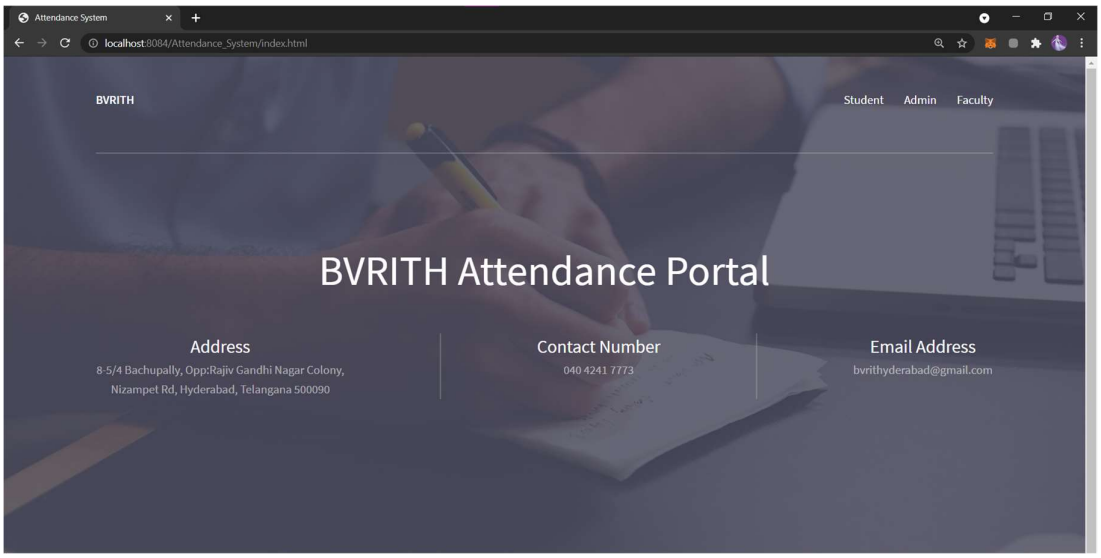

Fig 4.6.1: Test case showing the home page after pasting URL in browser

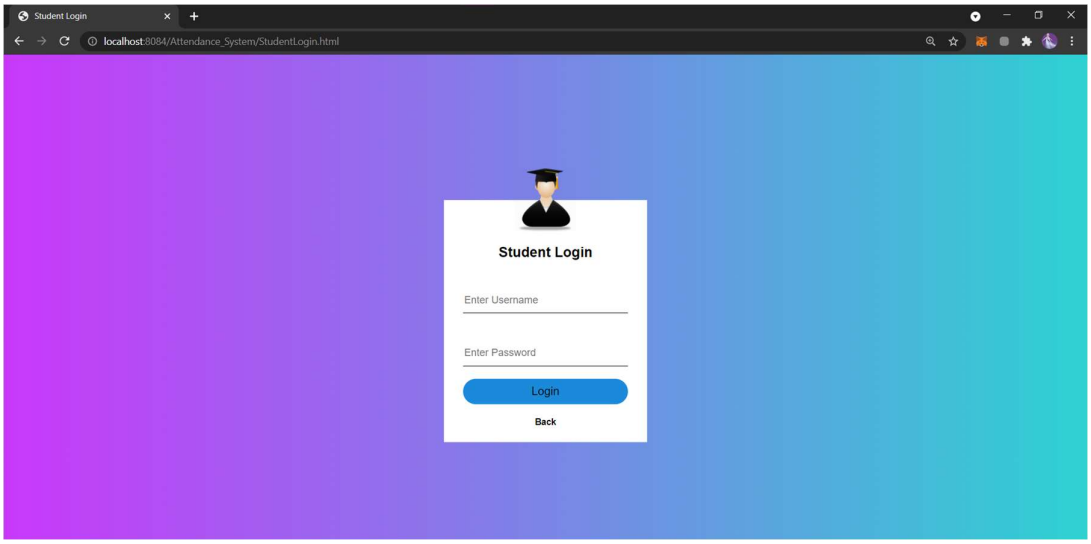

Fig 4.6.2: Test case showing navbar functionalities working

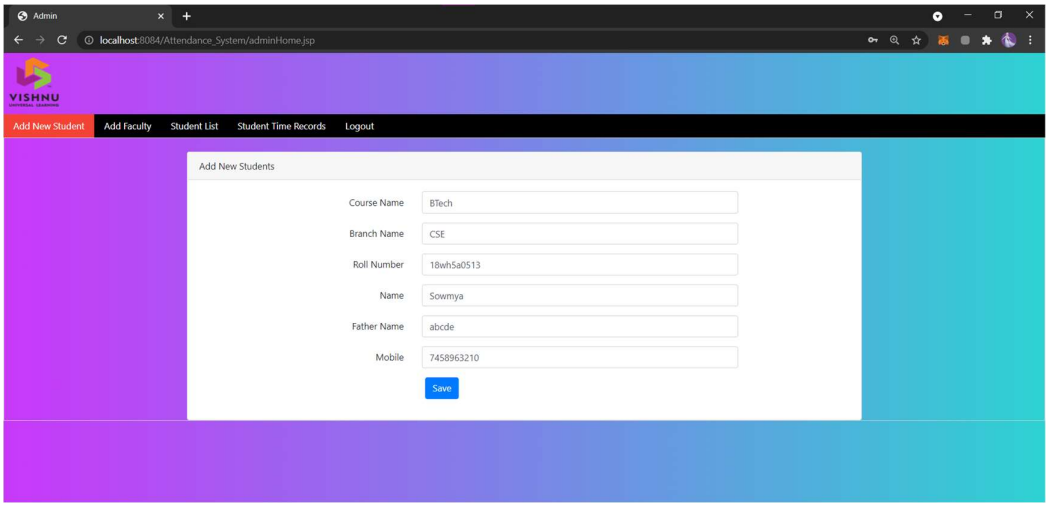

Fig 4.6.3: Test case showing login is done and navigated to home page

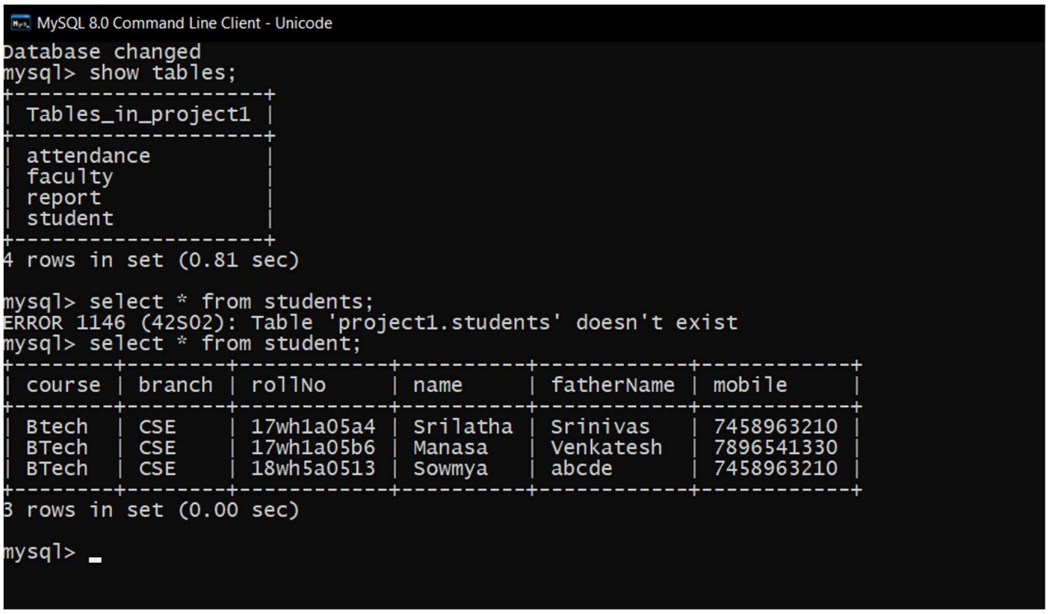

Fig 4.6.4: Test case showing student registration is working

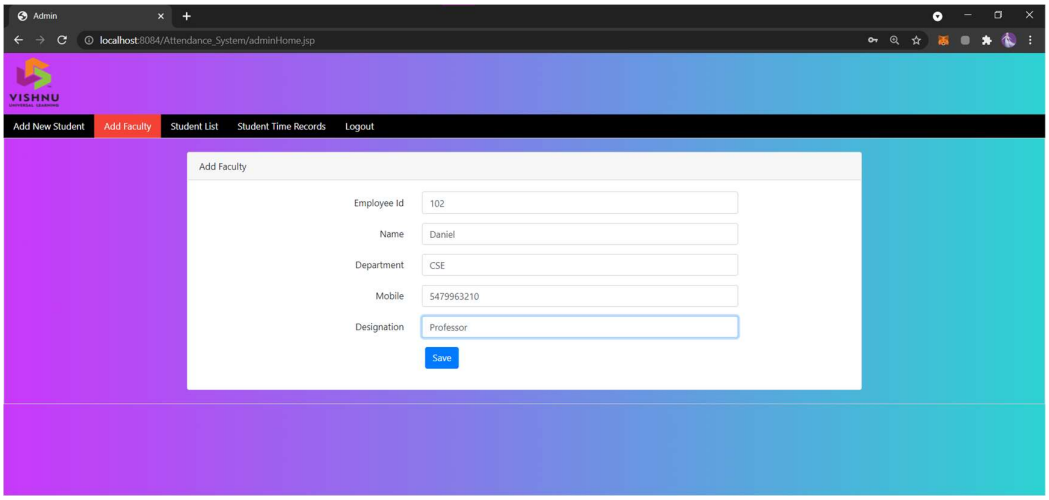

Fig 4.6.5: Test case showing faculty registration

```
mysql> select * from faculty;
+-------+---------+------------+------------+---------------------+
| empId | empName | department | mobile     | designation         |
+-------+---------+------------+------------+---------------------+
| 100   | Alex    | CSE        | 7896541230 | Assistant Professor |
| 102   | Daniel  | CSE        | 5479963210 | Professor           |
+-------+---------+------------+------------+---------------------+
2 rows in set (0.00 sec)

mysql>
```

Fig 4.6.6: Test case showing faculty registration is working

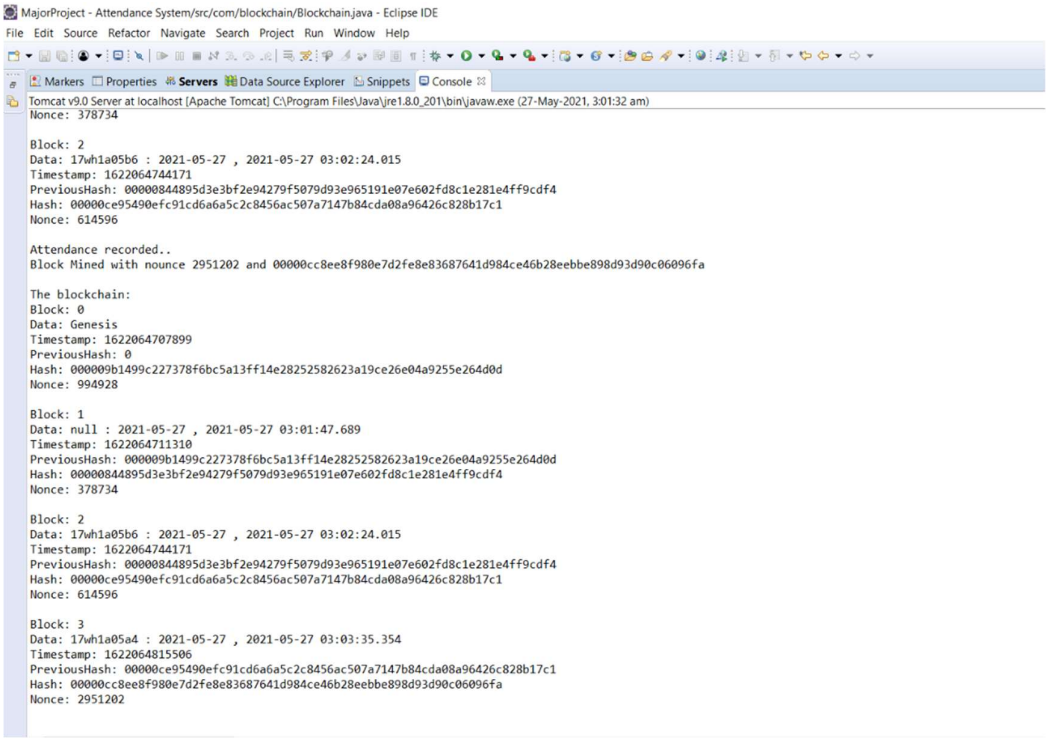

Fig 4.6.7: Test case showing attendance stored in blockchain

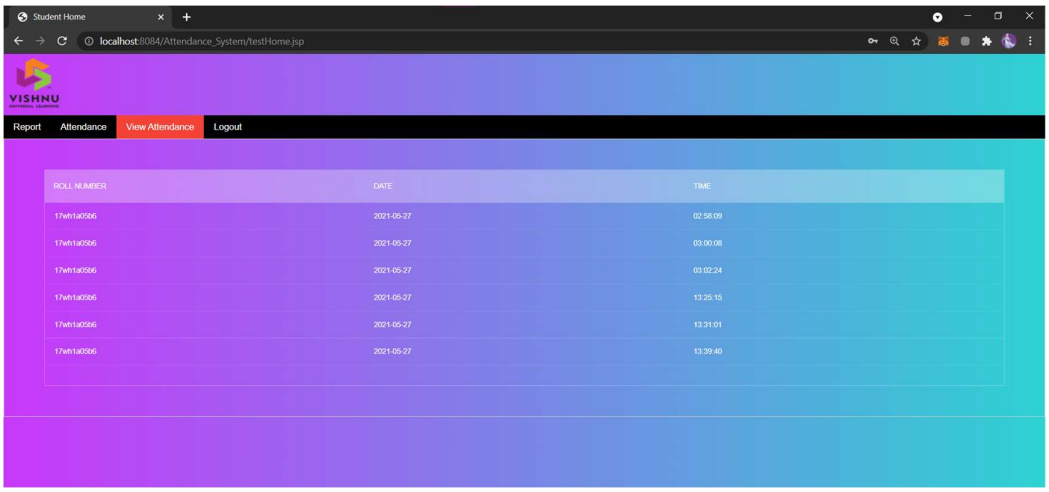

Fig 4.6.8: Test case showing student's attendance records

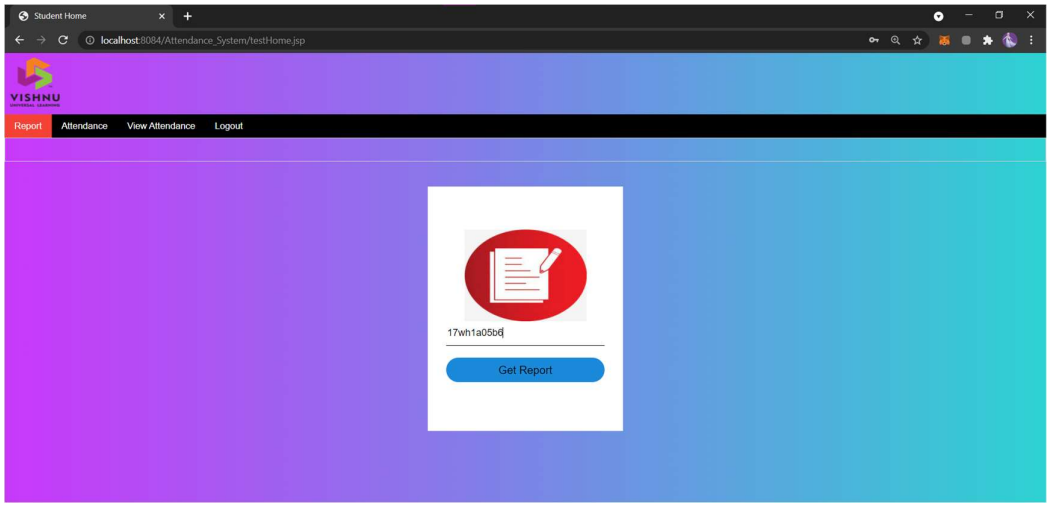

Fig 4.6.9: Test case to get student report

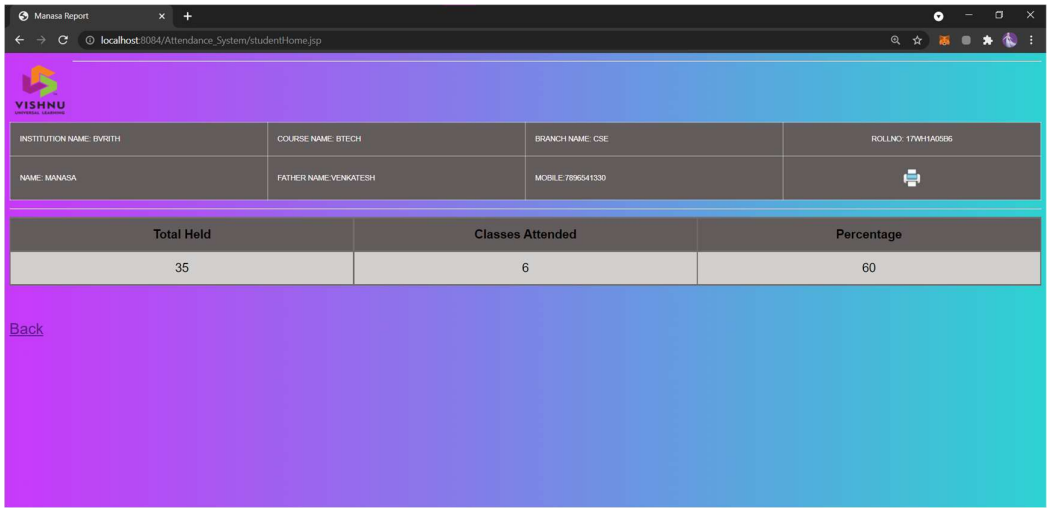

Fig 4.6.10: Test case showing Student's attendance report

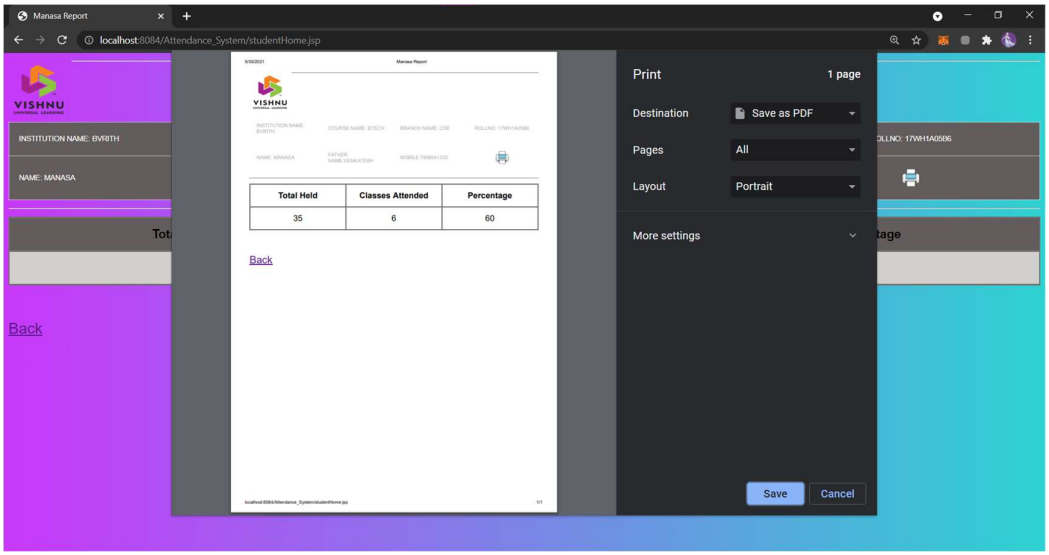

Fig 4.6.11: Test case showing download report is working

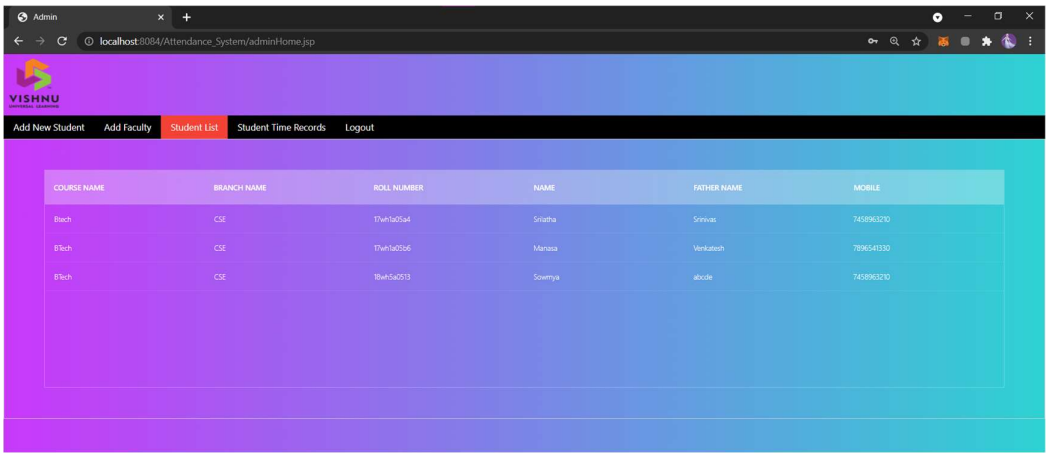

Fig 4.6.12: Test case showing all students details

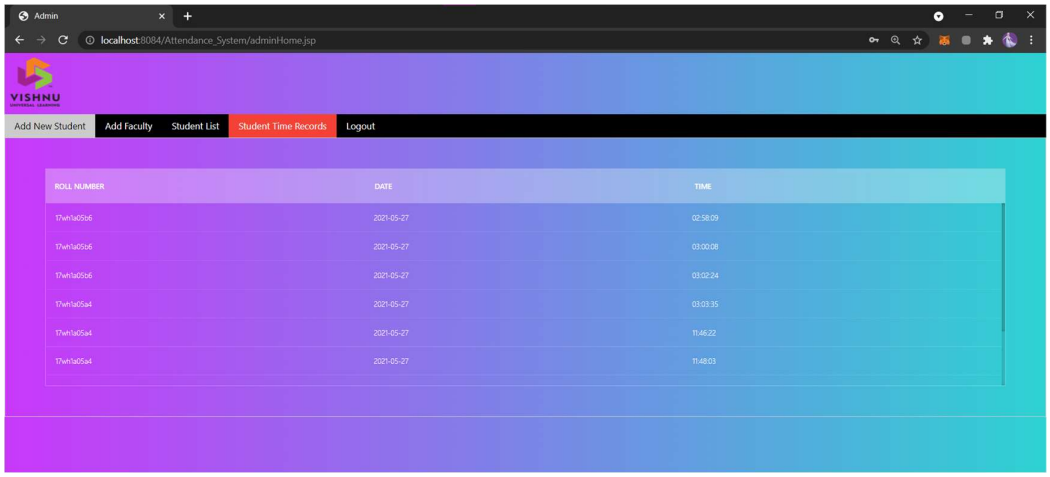

Fig 4.6.13: Test case showing all students attendance records

5. CONCLUSION AND FUTURE SCOPE

This application helps to automize the attendance system and makes easy to manage all the data. Encryption, decryption and blockchain makes the application very secure. The application has a very user-friendly UI and is made keeping UI and UX into consideration.

The future enhancement of this application is

- To use Ethereum to make the application up to data with the technologies
- To generate automatic weekly and monthly reports

6. REFERENCES

[1] K. Christidis and M. Devetsikiotis, “Blockchains and Smart Contracts for the Internet of Things”, in Digital Object Identifier 10.1109/Access.2016.2566339 Volume 4, 2016.

[2] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, and J.J. Kishigami, “Blockchain Contract: A Complete Consensus using Blockchain”, in IEEE International Conference on Consumer Electronics (ICCE), 2016.

[3] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosh, L. Bass, C. Pautassos, and P. Rimba, “A Taxonomy of Blockchain-Based Systems for Architecture Design”, in IEEE International Conference on Software Architecture, p243-252, 2017.

[4] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends:, IEEE 6th International Congress on Big Data, p. 557-564, 2017.

[5] M. Conoscenti, A.Vetro, and J.C. De Martin, “Blockchain for the Internet of Things: a Systematic Literature Review”, 978-1-5090-4320- 0/16, IEEE, 2016.

[6] J. Pereira, M.M. Tavalaei and H. Ozalp, “Blockchained-based platforms: decentralized infrasturctures and its boundary conditions”, Technological Forecasting & Social Change, pp. 94-102, 2019.