A REALTIME PROJECT REPORT ON

ATTENDANCE MANAGEMENT SYSTEM

A dissertation submitted in partial fulfilment of the

Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

Submitted by

(22B81A12A9)

(22B81A12B3)

(22B81A12C8)

Under the esteemed guidance of

J.Sadhana

Associate Professor, IT Department

CVR College of Engineering

DEPARTMENT OF INFORMATION TECHNOLOGY

CVR COLLEGE OF ENGINEERING

ACCREDITED BY NBA, AICTE & Affiliated to JNTU-H

Vastunagar, Mangalpally (V), Ibrahimpatnam (M), R.R. District, PIN-501 510

2023-2024

Cherabuddi Education Society's

CVR COLLEGE OF ENGINEERING

(An Autonomous Institution)

ACCREDITED BY NATIONAL BOARD OF ACCREDITATION, AICTE

(Approved by AICTE & Govt. of Telangana and Affiliated to JNT University)

Vastunagar, Mangalpalli (V), Ibrahimpatan (M), R.R. District, PIN - 501 510

Web : http://cvr.ac.in, email : info@cvr.ac.in

Ph : 08414 - 252222, 252369, Office Telefax : 252396, Principal : 252396 (O)

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the Project Report entitled **"Attendance Management System"** is a bonafide work done and submitted by T.Sai Dhruva(22B81A12A9), N.Sathvik(22B81A12B3) , T.Venu Gopal(22B81A12C8**)**, during the academic year 2023-2024, in partial fulfilment of requirement for the award of Bachelor of Technology degree in Information Technology from Jawaharlal Nehru Technological University Hyderabad, is a bonafide record of work carried out by them under my guidance and supervision.

Certified further that to the best of my knowledge, the work in this dissertation has not been submitted to any other institution for the award of any degree or diploma.

INTERNAL GUIDE
Dr.R.Seetharamaiah
Professor, IT Department

HEAD OF THE DEPARTMENT
Dr. Bipin Bihari Jayasingh
Professor, IT Department

PROJECT COORDINATOR
J. Sadhana
Assistant Professor, IT Department

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the project report entitled “**Attendance Management System**” is an original work done and submitted to the IT Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad in partial fulfilment of the requirement for the award of Bachelor of Technology in **Information Technology** and it is a record of bonafide project work carried out by us under the guidance of **Dr.R.Seetharamaiah, Department of Information Technology.**

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other Institute or University.

T. Sai Dhruva

(22B81A12A9)

T. Sathvik

(22B81A12B3)

T. Venu Gopal

(22B81A12C8)

ACKNOWLEDGEMENT

The satisfaction of completing this project would be incomplete without mentioning our gratitude towards all the people who have supported us. Constant guidance and encouragement have been instrumental in the completion of this project.

First and foremost, we thank the Chairman, Principal, and Vice Principal for availing infrastructural facilities to complete the major project in time.

We offer our sincere gratitude to our internal guide**, Dr.R.Seetharamaiah**, IT Department, CVR College of Engineering for his immense support, timely cooperation, and valuable advice throughout our project work.

We would like to thank the Professor in charge of Projects, **Dr.R.Seetharamaiah,** Professor, of Information Technology for his valuable suggestions in implementing the project.

We would like to thank the Head of Department, Professor **Dr. Bipin Bihari Jayasingh**, for his meticulous care and cooperation throughout the project work.

We also thank the **Project Review Committee Members** for their valuable suggestions.

ABSTRACT

ATTENDANCE MANAGEMENT SYSTEM

This project presents the development of a Attendance Management System (AMS) leveraging HTML, CSS, Java, and a Database Management System (DBMS). The AMS serves as a comprehensive platform for managing various aspects of attendance operations including attendance percentage, leave letter application, leave approval. By integrating frontend technologies with Java backend and DBMS, the system aims to streamline administrative tasks, enhance user experience, and promote efficient data management.

OBJECTIVES

- Design and implement an intuitive user interface using HTML and CSS, ensuring accessibility and visual appeal.
- Develop robust backend functionalities in Java.
- Establish a SQL database to store and manage student-related data, including student attendance, course details, and leave applications.
- Enable dynamic interaction between frontend and backend components, facilitating smooth data exchange and real-time updates.

METHODOLOGIES

- Frontend Development: Design user-friendly HTML pages with CSS styling to create an engaging and responsive user interface.
- Backend Development: Implement Java Servlets to handle user requests, process form submissions, and execute CRUD operations on the SQL database.
- Database Management: Design and configure a SQL database schema to efficiently store and retrieve student-related data.

CONCLUSION

The Attendance Management System developed in this project provides a robust solution for automating administrative tasks. By leveraging HTML, CSS, Java, and DBMS technologies, the system offers a user-friendly interface, efficient data management, and scalability for future enhancements. Overall, the AMS contributes to improving operational efficiency and enhancing the academic experience for students, faculty, and administrators.

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

An Attendance Management System is a software application used by colleges to manage student attendance efficiently. It helps administrators, teachers, and students to record, track, and manage attendance, leave applications, and approvals seamlessly.

Colleges have long been at the forefront of education and knowledge dissemination. However, traditional methods of managing student attendance have become increasingly inefficient in today's digital age. The advent of technology offers new opportunities to streamline attendance operations, improve accessibility, and enhance user experiences. This project aims to develop a robust Attendance Management System (AMS) that leverages modern technology to address the evolving needs of educational institutions in the 21st century.

This project is related to an Attendance Management System (AMS) explicitly designed for educational institutions such as universities or colleges where users include students, teachers, and administrators. The system utilizes HTML (Hypertext Markup Language) as the main programming language, along with JavaScript and CSS (Cascading Style Sheets). Tomcat is used as a web server to connect databases, and MySQL serves as the backend. This system can improve management and maintenance levels, making it easier for administrators and teachers to manage attendance data. Many institutions still manage attendance manually, requiring significant workforce and effort. With this system, the workload can be reduced, and it provides better online data storage solutions.

Educational institutions, traditionally seen as bastions of knowledge, are undergoing a profound transformation propelled by the digital revolution. This Attendance Management System (AMS) project represents a pivotal step towards modernizing attendance operations, aligning them with the fast-paced demands of contemporary academia. By harnessing the power of HTML, JavaScript, CSS, Tomcat, and MySQL, the system not only optimizes resource allocation but also enhances user engagement and accessibility. The traditional manual methods, labour-intensive and prone to human error, are being superseded by automated processes that streamline attendance tracking, leave applications, and approval tasks. With this innovative system, administrators and teachers can devote more time to educational activities and personalized support for students, rather than being bogged down by administrative burdens.

Moreover, by facilitating online data storage and management, the project heralds a new era of efficiency and scalability in attendance operations. As educational institutions embrace digital transformation, this project serves as a beacon of progress, symbolizing the evolution from antiquated attendance practices to a dynamic, tech-driven future. In essence, it's not just about managing attendance—it's about empowering educational institutions to efficiently track and manage student presence and fostering a vibrant learning community.

1.1 Motivation

The present study offers the intervention of a Management Information System (MIS) to the conventional attendance management. Colleges, as centres of learning, are experiencing unprecedented rates of change, both from internal and external environments. The new academic environment incorporates a changing student population, technology enhancements, transformation of administrative processes, digital information, new approaches to management, and a renewed commitment to planning and assessment throughout the organization.

One of the primary motivations behind developing our Attendance Management System (AMS) is to cater to the needs of colleges that prioritize monitoring students' attendance and leave activities. However, administrators and teachers have been slow to keep pace with this change. Our web application development in the attendance management domain is motivated by issues and needs that ensure the betterment of the student experience and the administration of work. Foremost among these is the call for a platform that is able to satisfy colleges on their need for effective monitoring of students' attendance and leave applications. This means that accessibility is highly enhanced, as the web application empowers students to connect to the system's resources from anywhere that has a network connection.

Digital transformation not only saves time but also enhances the visibility of attendance records, enabling students to seamlessly check their attendance status and apply for leave in real-time. In addition, features such as notifications and reminders contribute to the enhancement of organizational skills and accountability, ensuring students are up to date with their attendance responsibilities. Integration of the system into existing educational platforms enriches the experience and sense of community among students. Furthermore, data gathered through the web application offers valuable insights for making informed decisions in a way that shapes future strategies for managing attendance and academic performance.

The web has changed every aspect of life. The most visible change has occurred in the size, rate of change, and speed of information availability and delivery. The increasing expectation of users, changes in attendance tracking patterns brought about by increasing use of information and communication technologies, and preferences for online systems have resulted in automating attendance processes and services which underscore the adoption of the Attendance Management System. Attendance automation, which implies the application of information technologies to attendance operations and services with little supervision by people in developing countries, is limited and has been mainly adopted in academic institutions.

1.2 PROBLEM STATEMENT

Many colleges struggle with outdated and inefficient attendance management systems. These old systems make it hard for administrators and teachers to manage attendance records and provide good service to students. The existing systems often have confusing interfaces, limited features, and weak security. Colleges need a modern, easy-to-use Attendance Management System (AMS) that uses advanced technology like Java programming. This would automate tasks, streamline workflows, improve accessibility, and protect attendance records and user data.

This project aims to create a scalable, user-friendly, and feature-rich AMS that meets the changing needs of modern educational institutions. The system will allow students to apply for leave online and provide an efficient way for the heads of departments and teachers to approve these leave requests. The goal is to have a more efficient, accessible, and rewarding attendance management experience for staff and students, enhancing overall educational administration and student accountability.

1.3 PROJECT OBJECTIVES

The objectives of our Attendance Management System (AMS) project encompass various aspects aimed at improving attendance management, enhancing user experiences, and leveraging technology to meet the evolving needs of educational institutions. These objectives are outlined as follows:

Streamlining Attendance Tracking Processes

The primary goal of our AMS is to streamline attendance tracking processes to ensure efficient organization and accessibility of attendance records. This includes:

- Implementing a user-friendly interface for administrators and teachers to easily add, delete, and view attendance records.

- Enabling batch processing and bulk imports for expedited management of large student populations.

Facilitating Student Leave Applications

Another key objective is to facilitate seamless student leave application processes to enhance student engagement and ease of use. This involves:

- Providing intuitive forms for students to apply for leave with relevant details.

- Implementing secure authentication mechanisms such as username/password and email verification.

- Supporting integration with single sign-on (SSO) systems to enable students to access the attendance system using existing institutional credentials.

Enabling Approval Workflows for Leave Requests

Our AMS is designed to support efficient workflows for the approval of leave requests by heads of departments and teachers. This includes:

- Creating an easy-to-use interface for reviewing and approving or rejecting leave applications.

- Sending notifications to relevant parties to inform them of new leave requests and approval statuses.

- Allowing for customizable approval workflows to meet the specific needs and policies of different departments.

Supporting Integration and Customization

Our AMS aims to support seamless integration with existing academic systems and provide flexibility for customization to meet the specific needs and requirements of different colleges. This includes:

- Offering APIs (Application Programming Interfaces) and web services for integration with external systems such as learning management systems (LMS), student information systems (SIS), and authentication systems.

- Providing software development kits (SDKs) and documentation to facilitate customizations and extensions by developers and administrators.

- Collaborating with educational institutions to gather feedback and requirements for future enhancements and feature additions, ensuring continuous improvement and alignment with industry standards and best practices.

CHAPTER-2

LITERATURE REVIEW

2.1 EXISTING WORK

Attendance Management Systems (AMS) have evolved significantly, playing a pivotal role in automating attendance tracking, improving user experiences, and facilitating data-driven decision-making. This literature review highlights the benefits, key features, challenges, and emerging trends in AMS. It is evident that AMS are essential tools for modern educational institutions, and their continuous development will contribute to the growth and effectiveness of attendance management in the digital age. The data relating to the attendance and leave records of students is critical, as manual management can cause discrepancies during data encoding.

The students' attendance history is extensive and needs to be stored over a long period, requiring a substantial amount of files for storage. With this system, the level of data management can be significantly increased. Furthermore, administrators need to update attendance reports at any given time, which can be time-consuming when done manually. The system provides reports automatically and faster. This system helps administrators and teachers handle day-to-day attendance transactions. It also allows for the easy tracking of student attendance, leave applications, and approvals. For students, they can easily apply for leave and check their attendance status.

The importance of the project is to save time for both staff and students during the attendance and leave management processes. In addition, any issues regarding attendance can be quickly identified, and management can take timely action. This system can make attendance management more efficient and systematic.

An AMS increases reporting and monitoring rates. Updated records with the automatic attendance management system allow dynamic reporting and surveillance. This system enables administrators to access more accurate data and avoid any negligence regarding data storage. There are AMS in the market where users benefit from personalized notifications tailored to their attendance records, enhancing their academic experience. Our AMS uses a local database to store data, ensuring control over data storage and security without relying on cloud-based infrastructure. With a user-centric design, our interface ensures accessibility for all users.

Interoperability standards facilitate seamless integration with external systems, promoting resource sharing and collaboration among educational platforms. Moreover, our system champions digital record-keeping by maintaining accurate and secure attendance data. Advanced cybersecurity measures safeguard user data, ensuring privacy and confidentiality. Cost-effectiveness is paramount, with our solution minimizing operational expenses while maximizing efficiency. Continuous innovation drives our commitment to staying ahead of emerging trends and challenges, providing educational institutions with state-of-the-art tools and services.

The process phase of the Waterfall model is used to develop this system and is divided into separate phases; the result of one step will act as an input in the next phase sequentially. In addition, the Waterfall model will help plan and schedule system development. The development process moves from planning, analysing, designing, implementing, testing, and maintenance. Attendance Management Systems have evolved significantly, playing a pivotal role in automating attendance operations, improving user experiences, and facilitating data-driven decision-making. This literature review highlights the benefits, key features, challenges, and emerging trends in AMS. It is evident that AMS are essential tools for modern educational institutions, and their continuous development will contribute to the growth and effectiveness of attendance management in the digital age.

2.2 LIMITATIONS OF EXISTING WORK

1. Technology Stack Dependency:

Many existing Attendance Management System (AMS) solutions may be built on technologies that are less flexible or scalable compared to your Java-based system. This dependency on specific technology stacks could limit customization options or hinder integration with other systems and services.

2. Usability and Interface Design:

Some AMS solutions may lack a user-friendly interface or intuitive design, making it challenging for administrators, teachers, and students to navigate the system efficiently. Improving usability and interface design could enhance user satisfaction and productivity within the college environment.

3. Limited Functionality:

Certain AMS solutions may offer only basic functionalities for attendance tracking and leave management, without incorporating advanced features such as fine calculation, notifications, or comprehensive reporting capabilities. Your project's focus on comprehensive functionality could address these limitations and provide a more robust solution for attendance management.

4. Scalability Issues:

Existing AMS solutions may struggle to scale effectively to accommodate the growing needs of colleges with expanding student populations. By leveraging object-oriented programming principles and database management techniques, your project aims to offer a scalable and adaptable solution capable of meeting the evolving demands of modern educational institutions.

5. Integration Complexity:

Integrating the attendance management system with other institutional systems, such as student information systems or learning management systems, can be challenging due to compatibility issues and disparate data formats.

6. User Training Needs:

Transitioning from manual or outdated systems to a new attendance management system often requires extensive training for administrators, teachers, and staff to effectively utilize all features and functionalities.

7. Accessibility and Inclusivity:

Ensuring that the attendance management system is accessible to users with disabilities and meets diverse user needs, including multilingual support and interface customization options, presents ongoing challenges.

8. Security Concerns:

Safeguarding sensitive student data, including personal information and attendance records, against unauthorized access or data breaches is a paramount concern for attendance management systems, requiring robust security measures and compliance with data protection regulations.

9. Cost Considerations:

Implementing and maintaining a comprehensive attendance management system can incur significant upfront and ongoing costs, including software licenses, hardware infrastructure, and staff training, which may pose financial challenges for some educational institutions.

10. Remote Access and Connectivity:

Providing seamless access to attendance records and leave management services for remote users, including off-campus students and faculty, requires robust connectivity solutions and may pose technical challenges in areas with limited internet access or infrastructure.

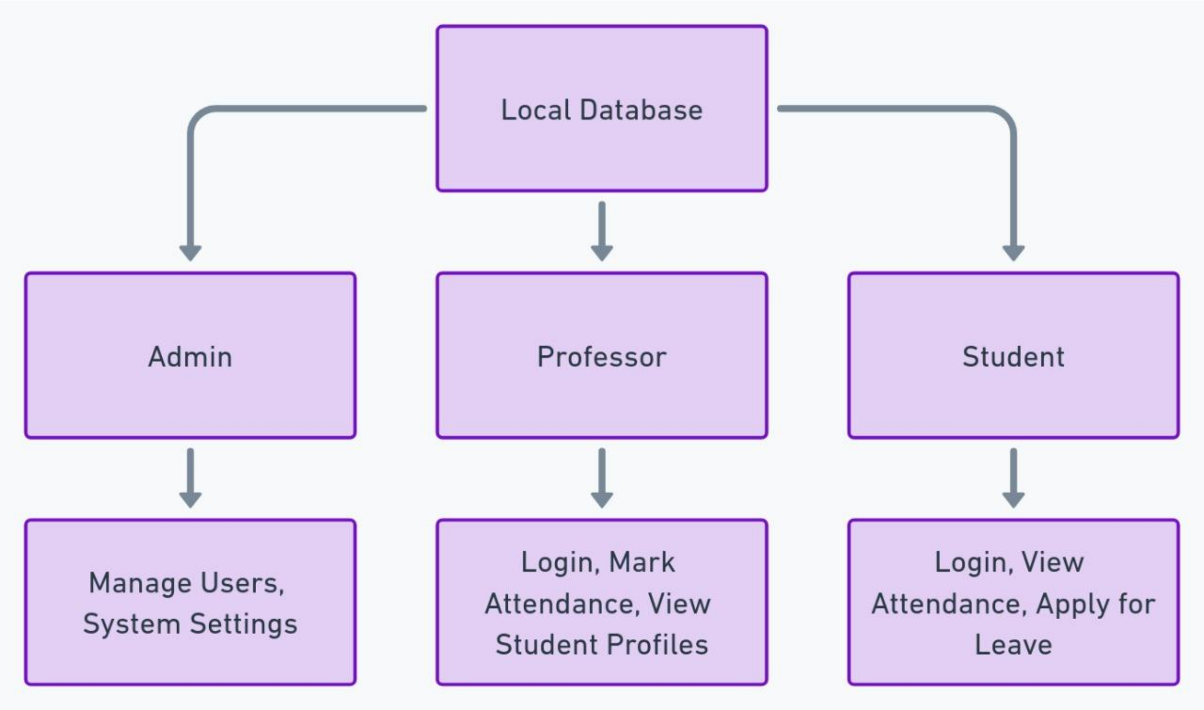
11. Customization and Adaptability:

Meeting the unique requirements and workflows of different types of educational institutions, while maintaining standardized and adaptable system architecture, can be a complex balancing act.

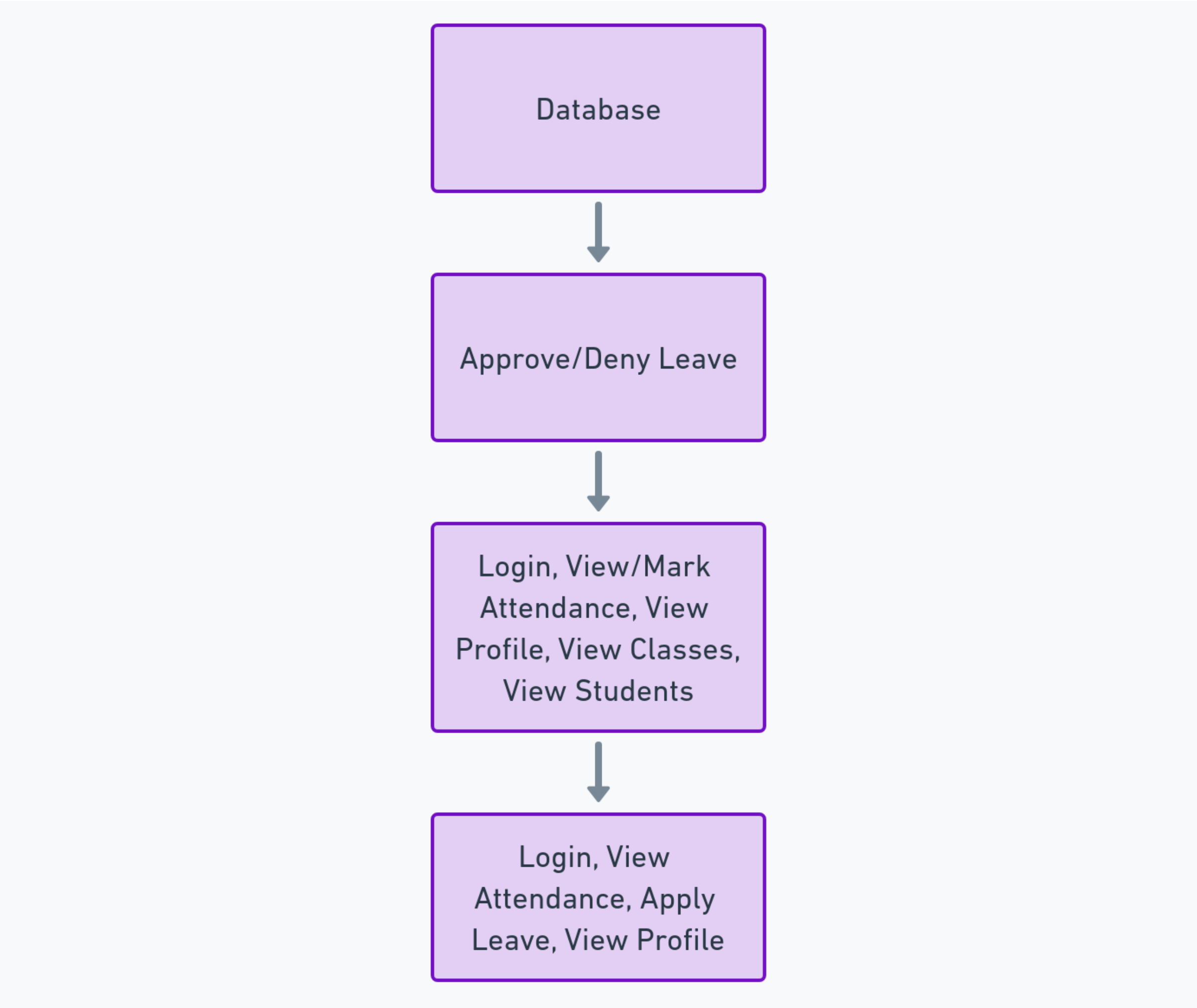
CHAPTER-3

DESIGN OR ARCHITECTURE

3.1 Architecture of Attendance Management System


Local Database
Admin
Professor
Student
Manage Users, System Settings
Login, Mark Attendance, View Student Profiles
Login, View Attendance, Apply for Leave

3.2 Design of Attendance Management System


Database
Approve/Deny Leave
Login, View/Mark Attendance, View Profile, View Classes, View Students
Login, View Attendance, Apply Leave, View Profile

CHAPTER-4

IMPLEMENTATION AND RESULTS

4.1 Implementation

Attendance Controller

```
package com.project.ams.controller;


import com.project.ams.request.MarkAttendanceRequest;
import com.project.ams.response.AttendanceResponse;
import com.project.ams.response.MarkAttendanceResponse;
import com.project.ams.response.StudentAttendanceResponse;
import com.project.ams.service.AttendanceService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

@RestController
@RequestMapping("/attendance")
public class AttendanceController {

    @Autowired
    AttendanceService attendanceService;
    @PostMapping("/mark")
    public MarkAttendanceResponse markAttendance(@RequestBody
MarkAttendanceRequest markAttendanceRequest) {
        return attendanceService.markAttendance(markAttendanceRequest);
    }

    @GetMapping("/Students")
    public AttendanceResponse getAttendanceReport(
            @RequestParam("subjectID") String subjectId,
            @RequestParam("minimumPercentage") double minimumPercentage) {

        return attendanceService.getAttendanceBySubject(subjectId);
    }

    @GetMapping("/{studentId}")
    public StudentAttendanceResponse getStudentAttendance(@PathVariable String
studentId){
        return attendanceService.getStudentAttendance(studentId);
    }
}
```

Attendance Entity

```
package com.project.ams.entity;

import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.PrePersist;

import java.util.UUID;

@Entity
public class Attendance {
    @Id
    private String id;

    @PrePersist
    public void ensureId() {
        if (this.id == null || this.id.isEmpty()) {
            this.id = UUID.randomUUID().toString();
        }
    }

    private double percentage;
    private int totalClasses;
    private int attendedClasses;
    private String studentId;
    private String subjectId;

    public Attendance() {
    }

    public Attendance( double percentage, int totalClasses, int
attendedClasses, String studentId, String subjectId) {

        this.percentage = percentage;
        this.totalClasses = totalClasses;
        this.attendedClasses = attendedClasses;
        this.studentId = studentId;
        this.subjectId = subjectId;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }


    public double getPercentage() {
        return percentage;
    }

    public void setPercentage(double percentage) {
        this.percentage = percentage;
    }

    public int getTotalClasses() {
        return totalClasses;
    }

    public void setTotalClasses(int totalClasses) {
        this.totalClasses = totalClasses;
    }

    public int getAttendedClasses() {
        return attendedClasses;
    }

    public void setAttendedClasses(int attendedClasses) {
        this.attendedClasses = attendedClasses;
    }

    public String getStudentId() {
        return studentId;
    }

    public void setStudentId(String studentId) {
        this.studentId = studentId;
    }

    public String getSubjectId() {
        return subjectId;
    }

    public void setSubjectId(String subjectId) {
        this.subjectId = subjectId;
    }
}
```

Attendance Repository

```
package com.project.ams.jpa;

import com.project.ams.entity.Attendance;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;

import java.util.List;
import java.util.Optional;

@Repository
public interface AttendanceRepository extends JpaRepository<Attendance,
String> {

    @Query("SELECT a from Attendance a where a.studentId = :studentId")
    Optional<Attendance> getAttendanceByStudentId(@Param("studentId") String
studentId);

    @Query("SELECT a from Attendance a where a.subjectId = :subjectId")
    List<Attendance> getAttendanceBySubject(@Param("subjectId") String
subjectId);

}
```

MarkAttendance Requets

```
package com.project.ams.request;

import com.project.ams.enums.AttendanceStatus;

import java.time.LocalDate;
import java.util.Map;

public class MarkAttendanceRequest {
    private LocalDate dateOfAttendance;
    private String sectionId;
    private String subjectId;

    private Map<String, AttendanceStatus> attendanceStatusMap;

    private String professorId;

    public Map<String, AttendanceStatus> getAttendanceStatusMap() {
        return attendanceStatusMap;
    }

    public void setAttendanceStatusMap(Map<String, AttendanceStatus>
attendanceStatusMap) {
        this.attendanceStatusMap = attendanceStatusMap;
    }

    public LocalDate getDateOfAttendance() {
        return dateOfAttendance;
    }

    public void setDateOfAttendance(LocalDate dateOfAttendance) {
        this.dateOfAttendance = dateOfAttendance;
    }

    public String getSectionId() {
        return sectionId;
    }

    public void setSectionId(String sectionId) {
        this.sectionId = sectionId;
    }

    public String getSubjectId() {
        return subjectId;
    }

    public void setSubjectId(String subjectId) {
        this.subjectId = subjectId;
    }

    public String getProfessorId() {
        return professorId;
    }

    public void setProfessorId(String professorId) {
        this.professorId = professorId;
    }


}
```

Attendance Response

```
package com.project.ams.response;

import java.util.Map;
```

```java
public class AttendanceResponse {
    private Map<String, Double> attendanceMap;

    public AttendanceResponse(Map<String, Double> attendanceMap) {
        this.attendanceMap = attendanceMap;
    }

    public Map<String, Double> getAttendanceMap() {
        return attendanceMap;
    }

    public void setAttendanceMap(Map<String, Double> attendanceMap) {
        this.attendanceMap = attendanceMap;
    }
}
```

Attendance Service (Main logic)

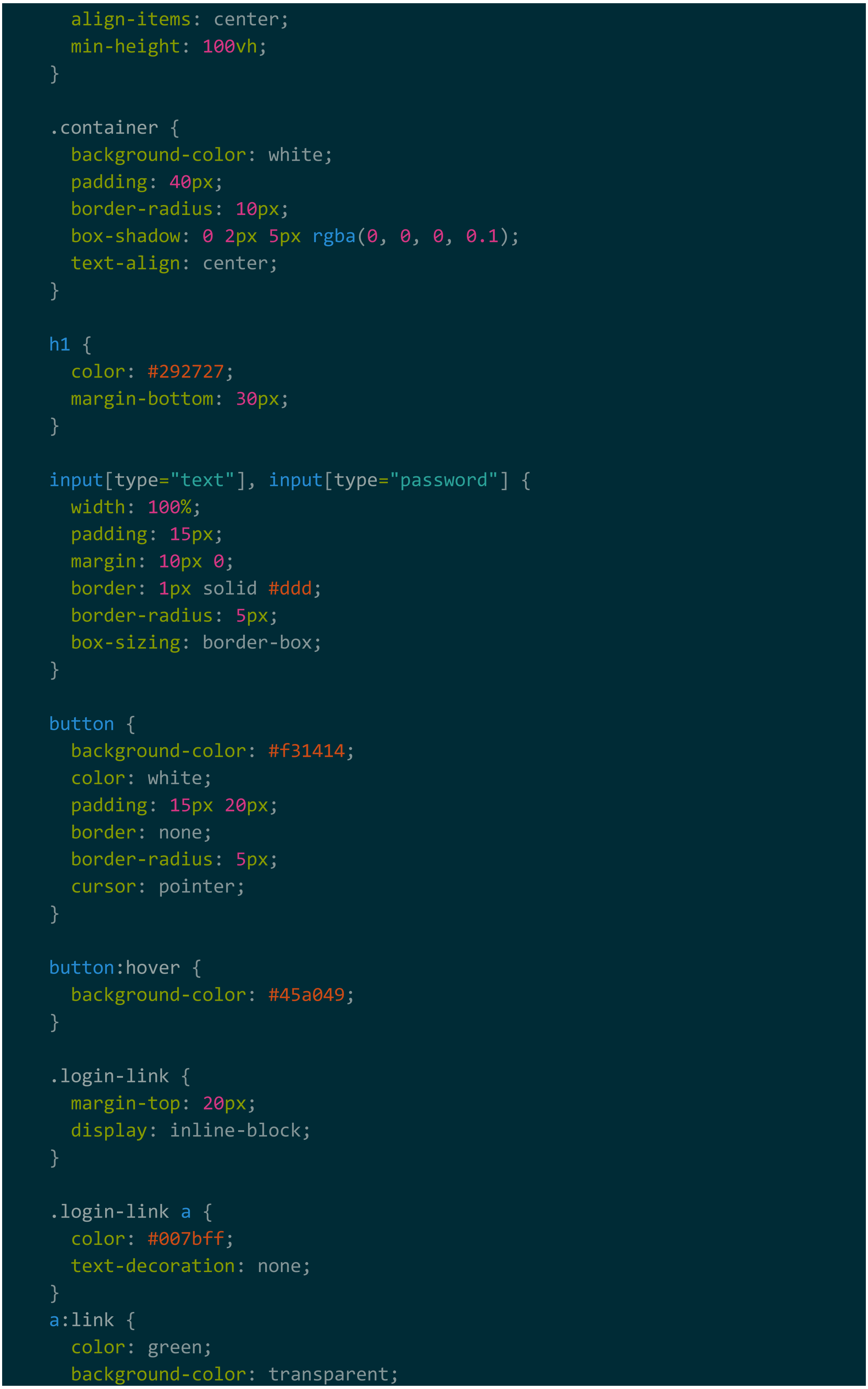
```java
package com.project.ams.service;

import com.project.ams.entity.Attendance;
import com.project.ams.entity.AttendanceRegister;
import com.project.ams.enums.AttendanceStatus;
import com.project.ams.jpa.AttendanceRegisterRepository;
import com.project.ams.jpa.AttendanceRepository;
import com.project.ams.request.MarkAttendanceRequest;
import com.project.ams.response.AttendanceResponse;
import com.project.ams.response.MarkAttendanceResponse;
import com.project.ams.response.StudentAttendanceResponse;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Optional;

@Service
public class AttendanceService {

    @Autowired
    AttendanceRegisterRepository attendanceRegisterRepository;
    @Autowired
    AttendanceRepository attendanceRepository;
    public AttendanceResponse getAttendanceBySubject(String subjectId) {
        List<Attendance> attendanceList =
attendanceRepository.getAttendanceBySubject(subjectId);
        Map<String, Double> studentAttendanceMap = new HashMap<>();
        for (Attendance attendance : attendanceList) {
            studentAttendanceMap.put(attendance.getStudentId(),
attendance.getPercentage());
        }
        return new AttendanceResponse(studentAttendanceMap);
    }

    public MarkAttendanceResponse markAttendance(MarkAttendanceRequest
markAttendanceRequest) {

        Map<String, AttendanceStatus> studentsAttendance =
markAttendanceRequest.getAttendanceStatusMap();

        for (Map.Entry<String, AttendanceStatus> studentAttendance :
studentsAttendance.entrySet()) {
            String studentId = studentAttendance.getKey();
            String attendanceStatus = studentAttendance.getValue().toString();

            // update in register
            updateInRegister(markAttendanceRequest, studentId,
attendanceStatus);

            // update in user
            updateInUser(markAttendanceRequest, studentId, attendanceStatus);
        }

        return new MarkAttendanceResponse(" 200 ");
    }

    private void updateInUser(MarkAttendanceRequest markAttendanceRequest,
String studentId, String attendanceStatus) {
        Optional<Attendance> attendance =
attendanceRepository.getAttendanceByStudentId(studentId);
        if (attendance.isPresent()) {
            Attendance oldAttendance = attendance.get();
            int attendedClasses = "PRESENT".equalsIgnoreCase(attendanceStatus)
? oldAttendance.getAttendedClasses() + 1 : oldAttendance.getAttendedClasses();
            int totalClasses = oldAttendance.getTotalClasses() + 1;
            oldAttendance.setTotalClasses(totalClasses);
            oldAttendance.setAttendedClasses(attendedClasses);
            oldAttendance.setPercentage(((double) attendedClasses /
totalClasses) * 100);
            oldAttendance.setSubjectId(markAttendanceRequest.getSubjectId());
            attendanceRepository.save(oldAttendance);
        } else {
            int totalClasses = 1;
            int attendedClasses = "PRESENT".equalsIgnoreCase(attendanceStatus)
? 1 : 0;
            double percentage = ((double) attendedClasses / totalClasses) *
100;
            Attendance newAttendance = new Attendance(percentage,
totalClasses, attendedClasses, studentId,
markAttendanceRequest.getSubjectId());
            attendanceRepository.save(newAttendance);
        }
    }

    private void updateInRegister(MarkAttendanceRequest markAttendanceRequest,
String studentId, String attendanceStatus) {
        AttendanceRegister register = new AttendanceRegister();
        register.setStudentId(studentId);
        register.setSectionId(markAttendanceRequest.getSectionId());
        register.setProfessorId(markAttendanceRequest.getProfessorId());
        register.setDateOfAttendance(markAttendanceRequest.getDateOfAttendance
());
        register.setAttendanceStatus(attendanceStatus);
        attendanceRegisterRepository.save(register);

    }

    public StudentAttendanceResponse getStudentAttendance(String studentId) {
        StudentAttendanceResponse studentAttendanceResponse = new
StudentAttendanceResponse();
        Optional<Attendance> attendance =
attendanceRepository.getAttendanceByStudentId(studentId);
        if (attendance.isEmpty()) {
            return studentAttendanceResponse;
        } else {
            Attendance studentAttendance = attendance.get();
            studentAttendanceResponse.setStudentId(studentId);
            studentAttendanceResponse.setPercentage(studentAttendance.getPerce
ntage());
            studentAttendanceResponse.setAttendedClasses(studentAttendance.get
AttendedClasses());
            studentAttendanceResponse.setTotalClasses(studentAttendance.getTot
alClasses());
            studentAttendance.setSubjectId(studentAttendance.getSubjectId());
            return studentAttendanceResponse;
        }
    }


}
```

Student.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>New Student</title>
</head>
<body>
<form action="#" th:action="@{/profile/student}" th:object="${student}"
method="get">
    <div>
        <label for="name">Name:</label>
        <input type="text" id="name" th:field="*{name}" />
    </div>
    <div>
        <label for="email">Email:</label>
        <input type="email" id="email" th:field="*{email}" />
    </div>
    <div>
        <label for="id">Id:</label>
        <input type="id" id="id" th:field="*{id}" />
    </div>
    <div>
        <label for="phonenumber">PhoneNumber:</label>
        <input type="phonenumber" id="phonenumber" th:field="*{phonenumber}"
/>
    </div>

    <div>
        <button type="submit">Submit</button>
    </div>
</form>

</body>
</html>
```

SignUp.html

```
<!DOCTYPE html>

<html>
<head>
  <title>Sign Up</title>
  <style>
    body {
      font-family: sans-serif;
      background-color: rgb(2, 113, 217);
      display: flex;
      justify-content: center;
```



```
  align-items: center;
  min-height: 100vh;
}

.container {
  background-color: white;
  padding: 40px;
  border-radius: 10px;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
  text-align: center;
}

h1 {
  color: #292727;
  margin-bottom: 30px;
}

input[type="text"], input[type="password"] {
  width: 100%;
  padding: 15px;
  margin: 10px 0;
  border: 1px solid #ddd;
  border-radius: 5px;
  box-sizing: border-box;
}

button {
  background-color: #f31414;
  color: white;
  padding: 15px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #45a049;
}

.login-link {
  margin-top: 20px;
  display: inline-block;
}

.login-link a {
  color: #007bff;
  text-decoration: none;
}
a:link {
  color: green;
  background-color: transparent;
```

```
      text-decoration: none;
    }

    a:visited {
      color: rgb(246, 10, 6);
      background-color: transparent;
      text-decoration: none;
    }

    a:hover {
      color: rgb(31, 79, 224);
      background-color: transparent;
      text-decoration: underline;
    }

  </style>
</head>
<body>
  <div class="container">
    <a href="https://cvr.ac.in/home4/"><img
src="https://www.careerguide.com/career/wp-content/uploads/2023/09/cvr-
placements.png" width="250"></a>
    <h1>SIGN UP</h1>
    <input type="text" placeholder="NAME" required style="width: 600px;"><br>
    <input type="text" placeholder="ID" required style="width: 600px;"><br>
    <input type="text" placeholder="E-MAIL" required style="width:
600px;"><br>

    <input type="password" placeholder="PASSWORD" required style="width:
600px;"><br>
    <input type="password" placeholder="CONFIRM PASSWORD" required
style="width: 600px;"><br>
    <button>Sign Up</button><br>
    <div class="login-link">
      Already have an Account? <a href="Login.html">Login</a>
    </div>
  </div>
</body>
</html>
```

Login.html

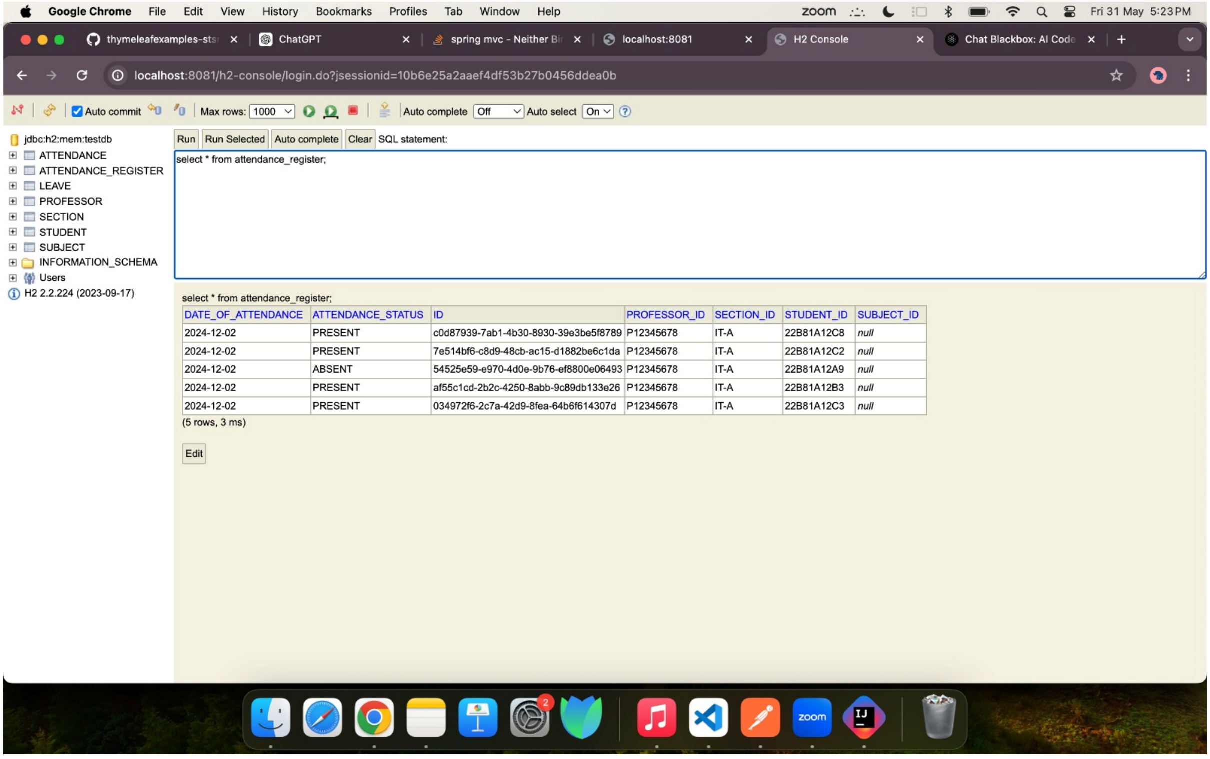
```
<!DOCTYPE html>

<html>
<head>
  <title>Login</title>
  <style>
    body {
```

```
  font-family: sans-serif;
  background-color: rgb(2, 113, 217);
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
}

.container {
  background-color: white;
  padding: 40px;
  border-radius: 10px;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
  text-align: center;
}

h1 {
  color: #333;
  margin-bottom: 30px;
}

input[type="text"], input[type="password"] {
  width: 100%;
  padding: 15px;
  margin: 10px 0;
  border: 1px solid #ddd;
  border-radius: 5px;
  box-sizing: border-box;
}

button {
  background-color: #f31414;
  color: white;
  padding: 15px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #45a049;
}

.login-link {
  margin-top: 20px;
  display: inline-block;
}

.login-link a {
  color: #007bff;
  text-decoration: none;
```

```
    }
    a:link {
      color: green;
      background-color: transparent;
      text-decoration: none;
    }

    a:visited {
      color: rgb(246, 10, 6);
      background-color: transparent;
      text-decoration: none;
    }

    a:hover {
      color: rgb(31, 79, 224);
      background-color: transparent;
      text-decoration: underline;
    }
  </style>
</head>
<body>
  <div class="container">
    <a href="https://cvr.ac.in/home4/"><img src="https://www.careerguide.com/career/wp-content/uploads/2023/09/cvr-placements.png" width="250"></a>
    <h1>LOGIN</h1>
    <input type="text" placeholder="USER ID" required style="width: 600px;"><br>

    <input type="password" placeholder="PASSWORD" required style="width: 600px;"><br>
    <div class="forgot-password">
      <a href="FogotPassword.html">Forgot Password</a>
    </div><br><br>
    <div class="new-user">
        New user? <a href="Home.html">SIGN UP</a>
      </div>

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

4.2 Results

Attendance Controller



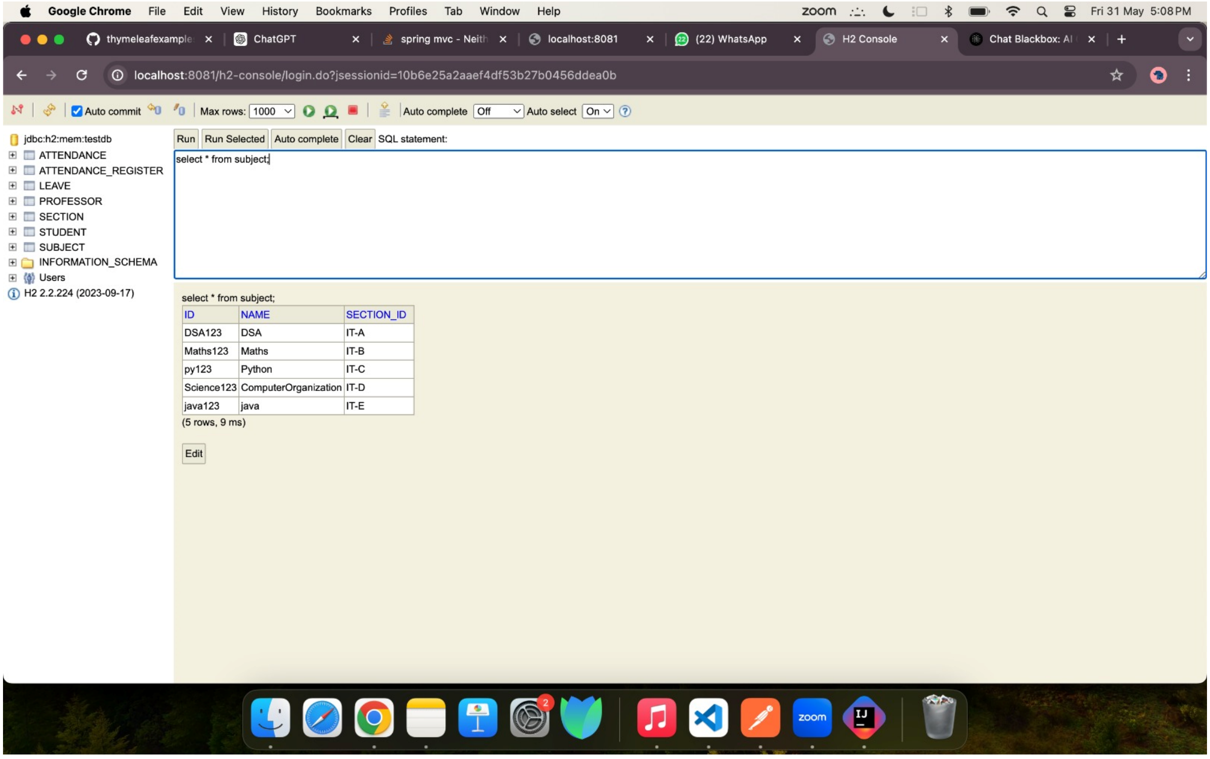
```
select * from attendance_register;
```

select * from attendance_register;

DATE_OF_ATTENDANCE	ATTENDANCE_STATUS	ID	PROFESSOR_ID	SECTION_ID	STUDENT_ID	SUBJECT_ID
2024-12-02	PRESENT	c0d87939-7ab1-4b30-8930-39e3be5f8789	P12345678	IT-A	22B81A12C8	*null*
2024-12-02	PRESENT	7e514bf6-c8d9-48cb-ac15-d1882be6c1da	P12345678	IT-A	22B81A12C2	*null*
2024-12-02	ABSENT	54525e59-e970-4d0e-9b76-ef8800e06493	P12345678	IT-A	22B81A12A9	*null*
2024-12-02	PRESENT	af55c1cd-2b2c-4250-8abb-9c89db133e26	P12345678	IT-A	22B81A12B3	*null*
2024-12-02	PRESENT	034972f6-2c7a-42d9-8fea-64b6f614307d	P12345678	IT-A	22B81A12C3	*null*

(5 rows, 3 ms)

Sections Database



☑ Auto commit, Max rows: 1000, Auto complete: Off, Auto select: On

SQL statement:

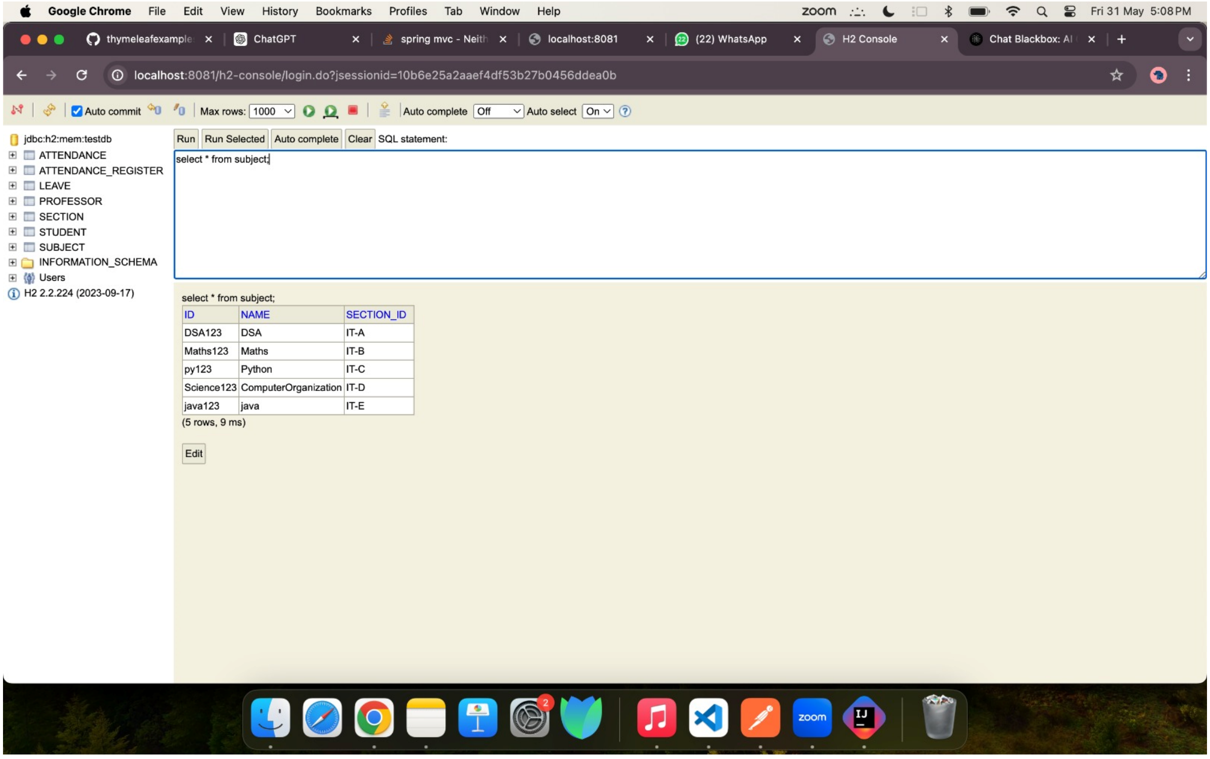
```
select * from subject;
```

select * from subject;

ID	NAME	SECTION_ID
DSA123	DSA	IT-A
Maths123	Maths	IT-B
py123	Python	IT-C
Science123	ComputerOrganization	IT-D
java123	java	IT-E

(5 rows, 9 ms)

Subjects Database


☑ Auto commit, Max rows: 1000, Auto complete: Off, Auto select: On

SQL statement:

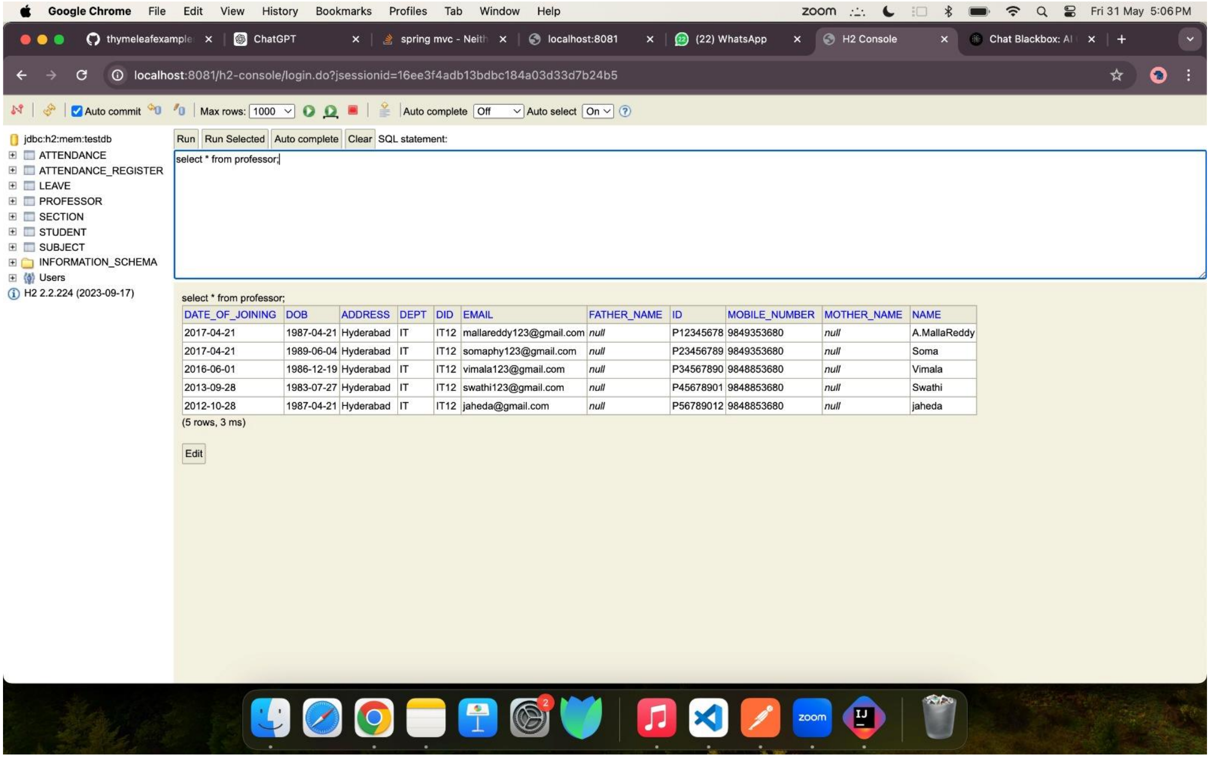
```
select * from subject;
```

select * from subject;

ID	NAME	SECTION_ID
DSA123	DSA	IT-A
Maths123	Maths	IT-B
py123	Python	IT-C
Science123	ComputerOrganization	IT-D
java123	java	IT-E

(5 rows, 9 ms)

Professor Database

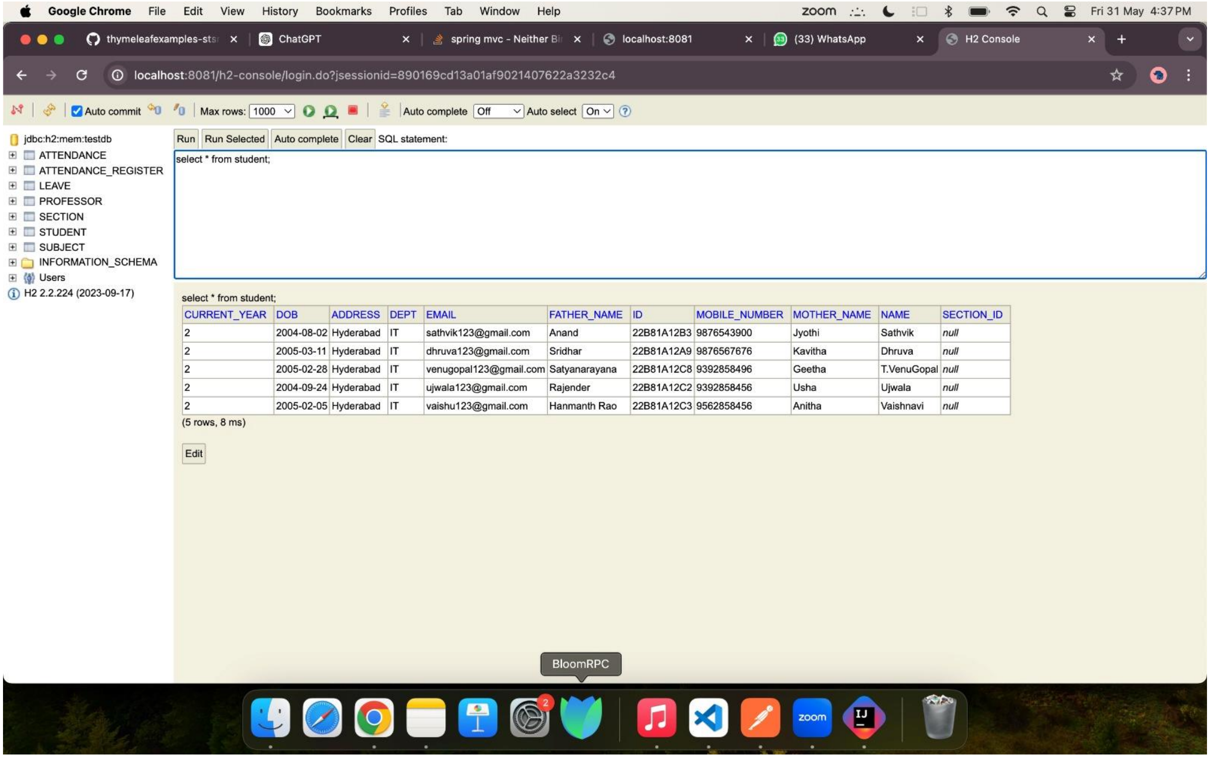


```
select * from professor;
```

DATE_OF_JOINING	DOB	ADDRESS	DEPT	DID	EMAIL	FATHER_NAME	ID	MOBILE_NUMBER	MOTHER_NAME	NAME
2017-04-21	1987-04-21	Hyderabad	IT	IT12	mallareddy123@gmail.com	*null*	P12345678	9849353680	*null*	A.MallaReddy
2017-04-21	1989-06-04	Hyderabad	IT	IT12	somaphy123@gmail.com	*null*	P23456789	9849353680	*null*	Soma
2016-06-01	1986-12-19	Hyderabad	IT	IT12	vimala123@gmail.com	*null*	P34567890	9848853680	*null*	Vimala
2013-09-28	1983-07-27	Hyderabad	IT	IT12	swathi123@gmail.com	*null*	P45678901	9848853680	*null*	Swathi
2012-10-28	1987-04-21	Hyderabad	IT	IT12	jaheda@gmail.com	*null*	P56789012	9848853680	*null*	jaheda

(5 rows, 3 ms)

Database(Student Table)

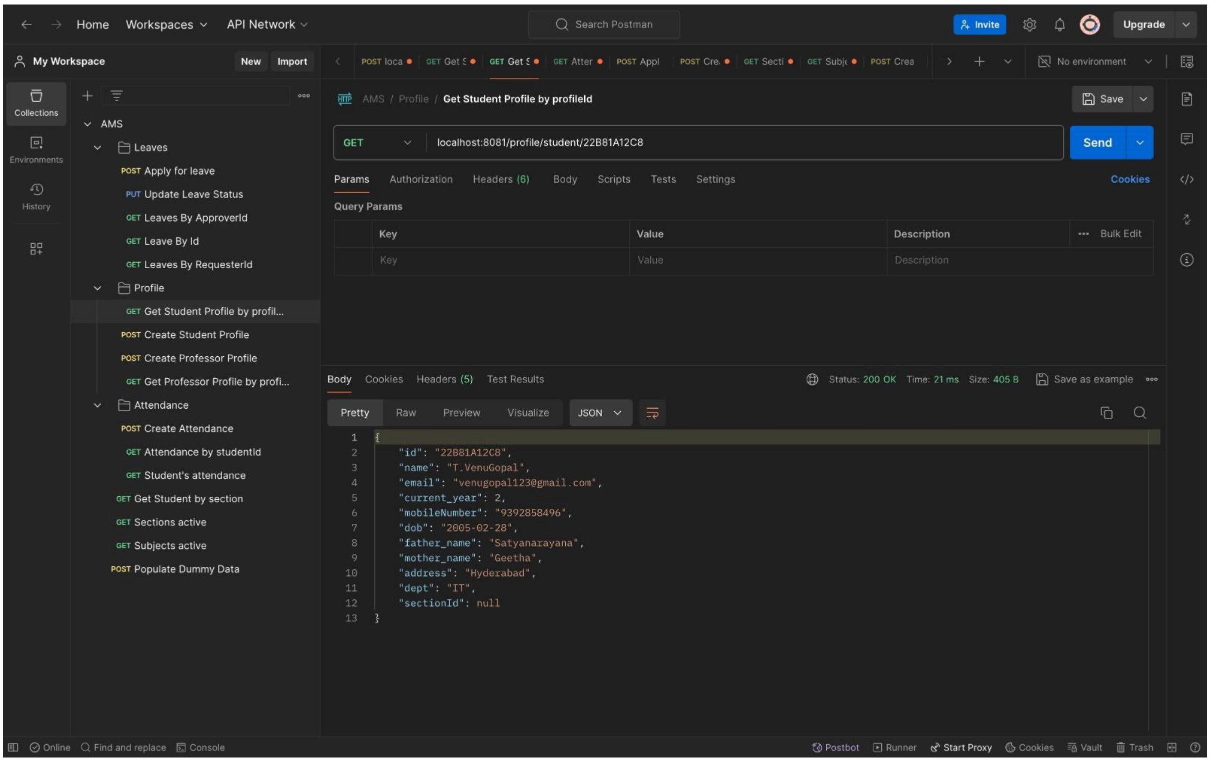


```
select * from student;
```

CURRENT_YEAR	DOB	ADDRESS	DEPT	EMAIL	FATHER_NAME	ID	MOBILE_NUMBER	MOTHER_NAME	NAME	SECTION_ID
2	2004-08-02	Hyderabad	IT	sathvik123@gmail.com	Anand	22B81A12B3	9876543900	Jyothi	Sathvik	*null*
2	2005-03-11	Hyderabad	IT	dhruva123@gmail.com	Sridhar	22B81A12A9	9876567676	Kavitha	Dhruva	*null*
2	2005-02-28	Hyderabad	IT	venugopal123@gmail.com	Satyanarayana	22B81A12C8	9392858496	Geetha	T.VenuGopal	*null*
2	2004-09-24	Hyderabad	IT	ujwala123@gmail.com	Rajender	22B81A12C2	9392858456	Usha	Ujwala	*null*
2	2005-02-05	Hyderabad	IT	vaishu123@gmail.com	Hanmanth Rao	22B81A12C3	9562858456	Anitha	Vaishnavi	*null*

(5 rows, 8 ms)

Postman-API Call Testing

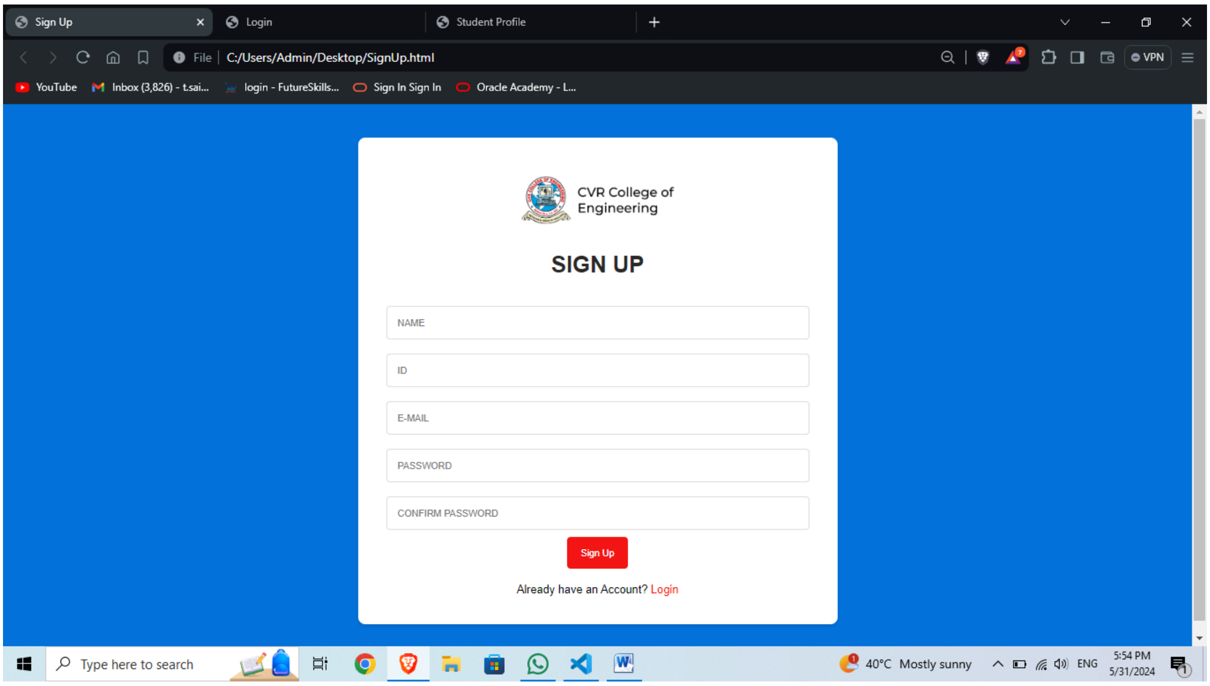


GET localhost:8081/profile/student/22B81A12C8

Status: 200 OK Time: 21 ms Size: 405 B

```
{
    "id": "22B81A12C8",
    "name": "T.VenuGopal",
    "email": "venugopal123@gmail.com",
    "current_year": 2,
    "mobileNumber": "9392858496",
    "dob": "2005-02-28",
    "father_name": "Satyanarayana",
    "mother_name": "Geetha",
    "address": "Hyderabad",
    "dept": "IT",
    "sectionId": null
}
```

SignUp Page

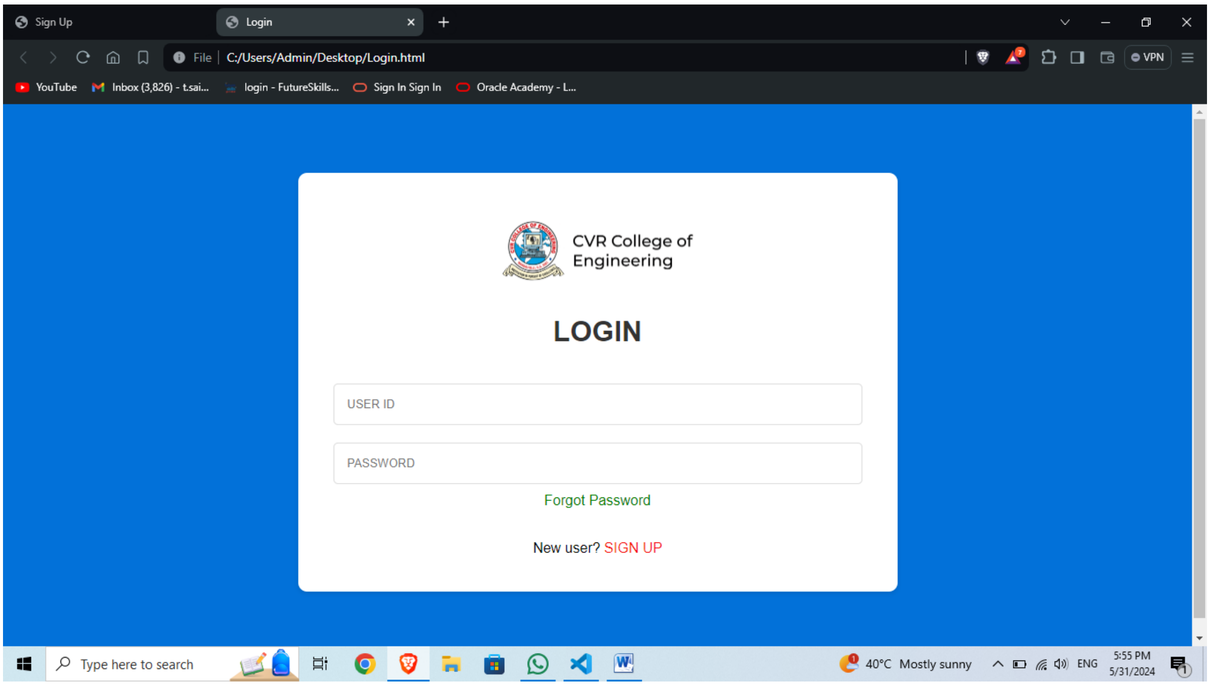


C:/Users/Admin/Desktop/SignUp.html

SIGN UP

Already have an Account? Login

Login Page



CONCLUSION

The implementation of an Attendance Management System (AMS) with a local database presents a significant enhancement in the management of student attendance within educational institutions. This system offers a user-friendly interface accessible to students, teachers, and Heads of Departments (HODs), facilitating seamless attendance tracking and leave management processes.

Firstly, this AMS streamlines attendance recording, replacing traditional manual methods prone to errors and time-consuming processes. With a local database, it ensures data integrity and security, allowing authorized users to record attendance with accuracy. Students can easily mark their attendance, while teachers and HODs can monitor and manage attendance records efficiently.

Moreover, the system includes a leave management feature, enabling students to apply for leave directly through the platform. This functionality simplifies the leave application process, eliminating paperwork and reducing administrative burden. HODs can then review and approve leave requests promptly, ensuring timely communication and decision-making.

Additionally, the AMS generates comprehensive reports and analytics, providing valuable insights into attendance patterns and trends. These insights empower educational institutions to identify areas for improvement and implement targeted interventions to enhance student engagement and academic performance.

Furthermore, the system promotes transparency and accountability by maintaining a detailed audit trail of attendance records and leave approvals. This transparency fosters trust among stakeholders and ensures compliance with institutional policies and regulations.

In conclusion, the implementation of an Attendance Management System with a local database offers numerous benefits for educational institutions. By automating attendance tracking, streamlining leave management processes, and providing valuable insights, the system enhances efficiency, transparency, and accountability. It empowers students, teachers, and HODs with tools to manage attendance effectively, ultimately contributing to improved academic outcomes and institutional excellence. As technology continues to evolve, further enhancements and integrations can be explored to optimize the functionality and usability of the system, ensuring its continued relevance and effectiveness in the dynamic landscape of education.

FUTURE ENHANCEMENTS

Several future enhancements could be considered to further improve the Attendance Management System with a local database:

1. Integration with Biometric or RFID Technology: Implementing biometric or RFID-based attendance tracking can enhance accuracy and security by eliminating the possibility of proxy attendance. Biometric systems could use fingerprint or facial recognition technology, while RFID tags or cards could be issued to students for automatic attendance recording.

2. Mobile Application: Developing a mobile application for the AMS would increase accessibility and convenience for users, allowing them to mark attendance, apply for leave, and access attendance records on their smartphones or tablets. This would particularly benefit students who are often on the move.

3. **Real-time Notifications:** Implementing real-time notifications through email or SMS would keep stakeholders informed about attendance-related activities, such as leave approvals, late arrivals, or absences. This proactive communication would improve transparency and facilitate prompt decision-making.

4. Predictive Analytics: Integrating predictive analytics capabilities into the system could help identify patterns and trends in attendance data, enabling early intervention for students at risk of poor attendance or academic performance. Predictive models could also forecast future attendance patterns based on historical data.

5. Customizable Reporting: Enhancing reporting capabilities with customizable templates and dashboards would allow administrators to generate tailored reports based on specific criteria, such as attendance by class, department, or time period. This would facilitate data-driven decision-making and performance monitoring.

6. Integration with Learning Management Systems (LMS): Integrating the AMS with existing LMS platforms would streamline data exchange and improve interoperability between attendance management and academic activities. This integration would provide a holistic view of student engagement and performance.

7. Machine Learning for Automated Decision-making: Leveraging machine learning algorithms for automated decision-making, such as predicting the likelihood of leave approval based on historical data and contextual factors, could expedite the leave management process while maintaining fairness and consistency.

8. Enhanced Security Measures: Implementing advanced security measures, such as encryption and multi-factor authentication, would safeguard sensitive attendance data from unauthorized access or cyber threats, ensuring compliance with data protection regulations.

By incorporating these future enhancements, the Attendance Management System can evolve into a more robust, user-friendly, and predictive tool for effectively managing attendance and promoting student success in educational institutions.

REFERENCES

1. Spring Boot: https://spring.io/projects/spring-boot
2. MavenRepositories: https://mvnrepository.com/search?q=spring+boot+dev+tools&ref=opensearch
3. Hibernate: https://hibernate.org/
4. W3schools: https://www.w3schools.com/html/