Abstract

Attendance is an important part of daily classroom ascertainment for the teacher for his or her smooth running of the class. At the beginning and end of the class, usually, the teacher checks the attendance, but the manual attendance system may lead to the teacher missing someone or some students may answer multiple times leading to a proxy attendee. Face recognition-based attendance system involves the process of recognizing faces for taking attendance by using face recognition technology based on visual and information technology.

First we encode a picture using the HOG algorithm to create a simplified version of the image. Using this simplified image, find the part of the image that most looks like a generic HOG encoding of a face. Then we manipulate the position of the image such that we get facial features centered and clearly. We pass the image through a neural network that knows how to measure features of the face. Save those 128 measurements. Then by comparing all the measurements stored we find the right match while we scan for faces.

A program to detect and recognize face of the students runs at the backend and marks the attendance on a csv file and inserts the name of the student and date into the respective ongoing subject's database. A system is maintained to view the attendance records. An admin can add teachers and students by logging into the system, to make sure that no access is given to strangers who are not enrolled in college. Once the hour is done an email is sent the respective subject teacher according to the time table that is fed into the system.

The noteworthy benefits would be to evade the tedious process of manually calling out attendance via roll call. This system also helps avoiding any possible error on the lecturer's part as well as eliminates proxy attendees. The system is also time saving and saves lecturer's energy. The proposed system is fully automated, managing the records and keeping a track of day-to-day activities will become much easier than the manual system.

TABLE OF CONTENTS

	Page Number
Acknowledgment	i
Abstract	ii
Table of Contents	iii
List of Tables	v
List of Figures	v
Chapter 1. Introduction	1
1.1 Background	1
1.2 Problem Statement	1
1.3 Scope and Importance	2
Chapter 2. Software Requirement Specification	3
2.1 Functional requirements	3
2.2 Software requirements	3
2.3 Hardware requirements	3
2.4 Non-Functional Requirements	3
Chapter 3. System Design	5
3.1 Abstract Design	5
3.1.1 Architecture Diagram	5
3.1.2 Use Case Diagram	6
3.2 Functional Design	7
3.2.1 Modular Design diagram	7
3.2.2 Sequence diagram	8
3.3 Control Flow Design	10
3.3.1 Algorithm for logic implementation	10
3.4 Access layer design	11
3.4.1 Database Schema	11
3.5 Presentation Layer Design	12
3.5.1 User Interface flow design	12
Chapter 4. Implementation	13
4.1 Software and Hardware requirements	13
4.1.1 Hardware Requirements	13

4.1.2 Software requirements 13
4.2 Pseudo Code 15

Chapter 5. Testing 17
5.1 Test Objectives 17
5.2 Types of Testing Conducted 17
5.2.1 Unit Testing 17
5.2.2 Integration Testing 20

Chapter 6. Conclusion and Future Work 22

Bibliography 23

Appendix-1: Screenshots of the Application 24

List of Tables

Table No.	Title	Page Number
5.1	Use cases for unit testing	18
5.2	Use cases for integration testing	21

List of Figures

Figure No.	Title	Page Number
3.1	Architecture diagram	05
3.2	Use case diagram	06
3.3	Modular design diagram	07
3.4	Sequence diagram of Automatic Attendance System	08
3.5	Sequence Diagram of Admin interacting with the Automatic Attendance system	08
3.6	Sequence diagram of a student interacting with Automatic Attendance System	09
3.7	Sequence diagram of teacher interacting with Automatic Attendance System	09
3.8	Flowchart of HOG algorithm	10
3.9	Schema diagram of the system	11
3.10	User interface flow diagram of Automatic Attendance system	12

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 Background

Attendance marking in a classroom during a lecture is not only an onerous task but also a time-consuming one at that. Due to an unusually high number of students present during the lecture, there will always be a probability of proxy attendances. Attendance marking with conventional methods has been an area of challenge. The growing need for efficient and automatic techniques for marking attendance is a growing challenge in the area of face recognition. In recent years, the problem of automatic attendance marking has been widely addressed through the use of standard biometrics like fingerprint and Radio frequency identification tags, etc., However, these techniques lack the element of reliability. In this proposed project an automated attendance marking and management system is proposed by making use of face detection and recognition algorithms. Instead of using conventional methods, this proposed system aims to develop an automated system that records the student's attendance by using facial recognition technology. The main objective of this work is to make the attendance marking and management system efficient, time-saving, simple, and easy. Here faces will be recognized using face recognition algorithms. The processed image will then be compared against the existing stored record and then attendance is marked in the database accordingly. Compared to the existing system traditional attendance marking system, this system reduces the workload of people. This proposed system will be implemented in 4 phases Image Capturing, Segmentation of images and Face Detection, Face comparison and Recognition, and Updating the attendance in the database as well as on an excel sheet.

1.2 Problem Statement

Attendance is an integral part of daily classroom ascertainment for the teacher for his or her smooth running of the class. At the commencement and the end of the class, Usually, The lecturer calls out attendance via roll call, But this very manual process of calling out attendance is always prone to human error as it may result in the lecturer missing out on a students name or some students may answer multiple times leading to a proxy attendee.

A face recognition-based attendance system involves the process of recognizing faces for taking attendance by using face recognition technology based on high-definition monitor video and other information technology.

1.3 Scope and Importance

Almost all academic institutions require attendance records of students and maintaining attendance manually can be hectic as well as time-consuming. Hence maintaining attendance automatically with the help of face recognition will be very helpful and less prone to errors as compared to the manual process. This will also reduce the manipulation of attendance records done by students and it will save time as well.

Attendance marking becomes foolproof in nature, students cannot carry out the previous means of false proxies for their friends as the system needs the faces of the students. The face recognition-based attendance system would help save time that at moments can get lost due to students disrupting the normal attendance marking method.

SOFTWARE REQUIREMENT SPECIFICATION

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Functional requirements

Admin Module:

- Login page to authenticate as an admin
- To add students and faculty
- To add timetable

User Module (Student/Teacher):

- Login page to authenticate as a user

2.2 Software requirements

- Windows 8 or higher
- Microsoft excel
- Python 3.10
- The latest version of all the required libraries

2.3 Hardware requirements

- IP camera / Web camera
- Computer
- Processor: Intel® Core™ i3-7020U CPU @ 2.30GHz 2.30 GHz and above
- Installed RAM: 4GB (Minimum), 8GB (Recommended)

2.4 Non-Functional Requirements

- Availability: The application must be available at all times as long as it is in proper working condition.
- Accuracy: The application should be able to provide accurate results, as the faculty relies on the system.
- Portability: The application can be migrated to a computer/laptop with ease irrespective of the operating system used.

- Reliability: The system automatically takes the attendance of the class, immaterial of the conditions of the environment hence, being reliable.

SYSTEM DESIGN

CHAPTER 3

SYSTEM DESIGN

3.1 Abstract Design

3.1.1 Architecture diagram

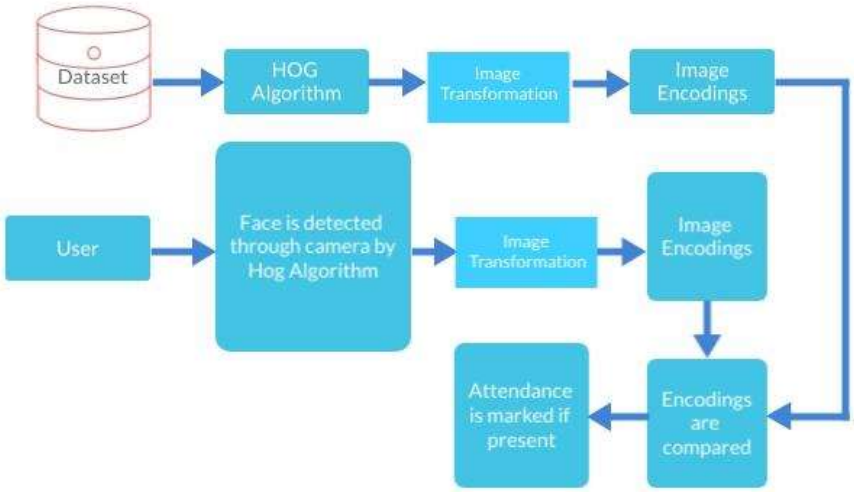

Fig. 3.1: Architecture Diagram

An architecture diagram is a graphical representation of a set of concepts, that are part of the architecture, including their principles, elements, and components. The diagram explains the system software in the perception of an overview of the system. Automatic Attendance System using Computer Vision is a Machine Learning-based application, designed to help the teaching faculty to take the attendance without pen and paper. The dataset consists of photographs of students. The HOG algorithm is applied to the dataset to perform image transformations like finding various gradients for each pixel of the image and centering the image in case the face of the student is not towards the center. Then, the image encodings are performed for the training images. When the user enters the environment, the camera takes frame by frame to detect and recognize the face of the students and then the image transformations, as well as the image encodings, are performed on the live images which are then compared with the encodings of the training images. If

the encodings are similar, then the face is detected and recognized and attendance is marked on the CSV file which later, is displayed in the user interface.

3.1.2 Use case diagram

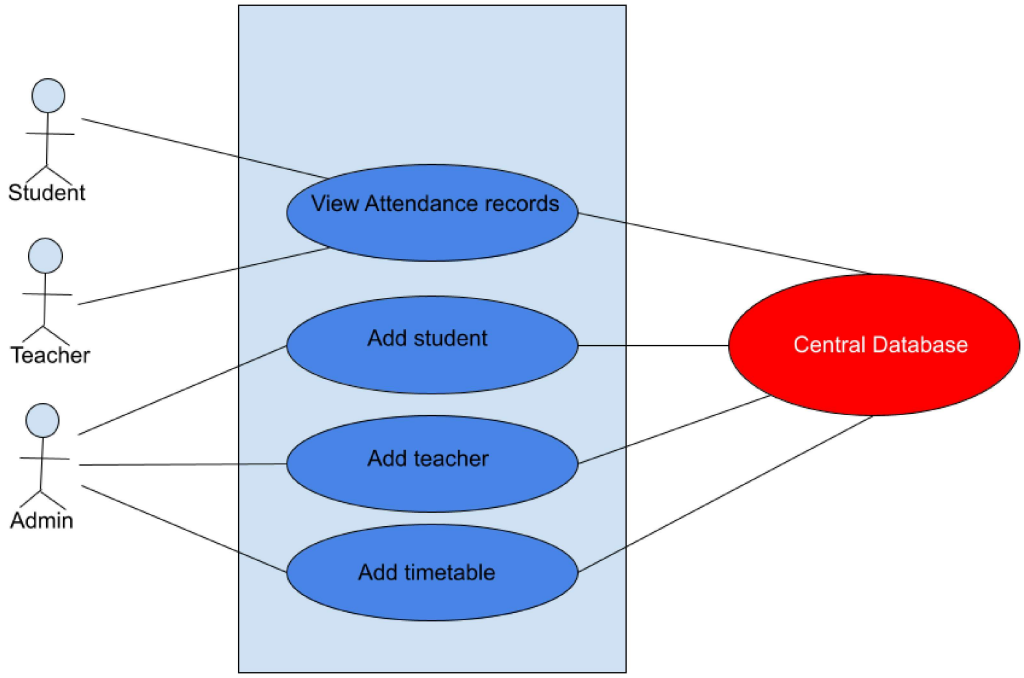

Fig. 3.2: Use case diagram

In the Unified Modelling Language (UML), a use case diagram summarizes the details of the users of the system and their interactions with the system. A set of specialized symbols and connectors are used to build it.

An effective use case diagram can help to represent:

- Scenarios in which the system or application interacts with people, organizations, or external systems.
- Goals that the system or application helps those entities achieve.
- The scope of the system.

The central database contains information about the students pertaining to the attendance of each student. The admin maintains the attendance system for smooth functioning. The students can view their respective attendance records and the faculty can view the status of

the student's attendance. The admin can add a student, teacher, and timetable so that the application can work according to the timetable.

3.2 Functional Design

3.2.1 Modular design diagram

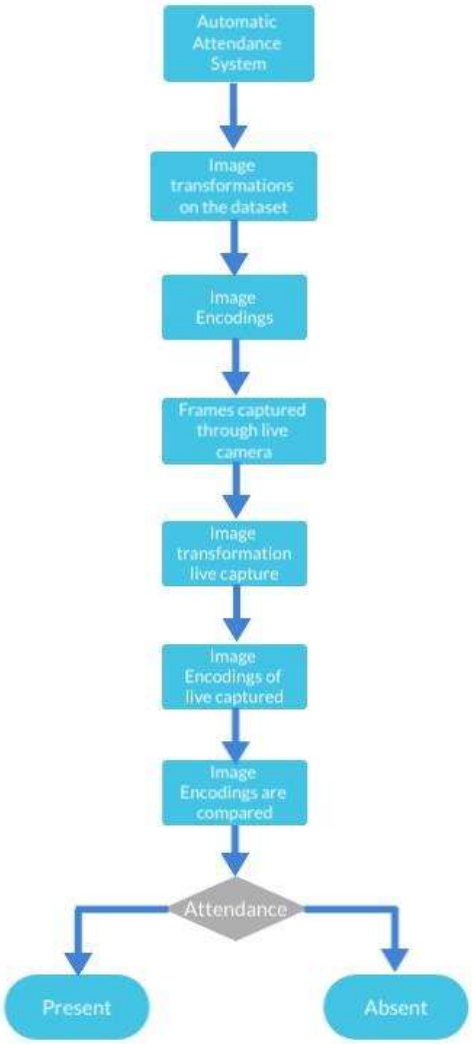

Fig. 3.3: Modular design diagram of Automatic Attendance System

The above modular design diagram shows the various modules available in the automatic attendance system. Image transformation is performed on the dataset and then its image encodings are measured. After this step, frames are captured through the camera live and image transformation is performed on these frames such as rotation, shearing, etc. The image encodings of the training dataset and the captured frames will be compared in order to detect and recognize the face and hence attendance will be marked.

3.2.2 Sequence diagram

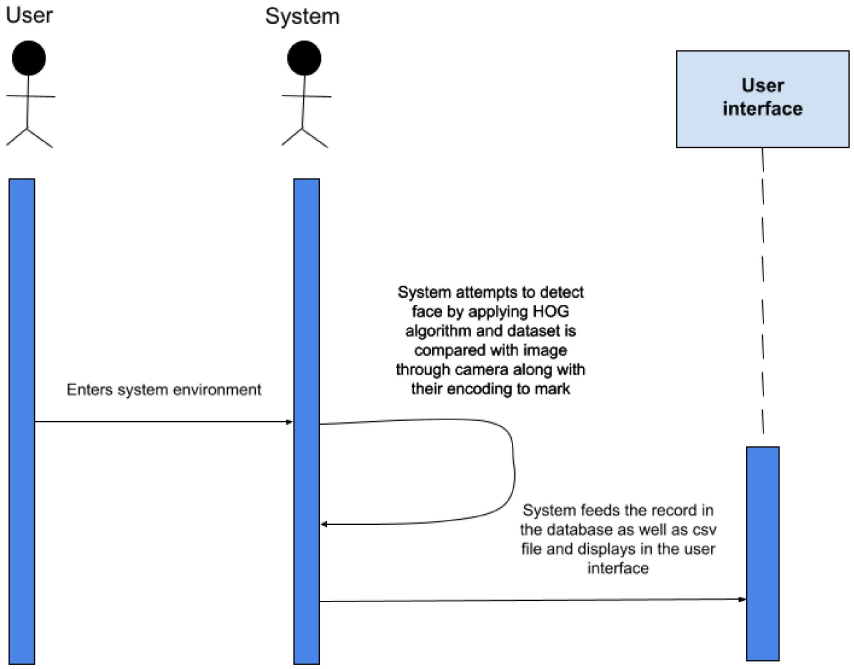

Fig. 3.4: Sequence Diagram of Automatic Attendance system

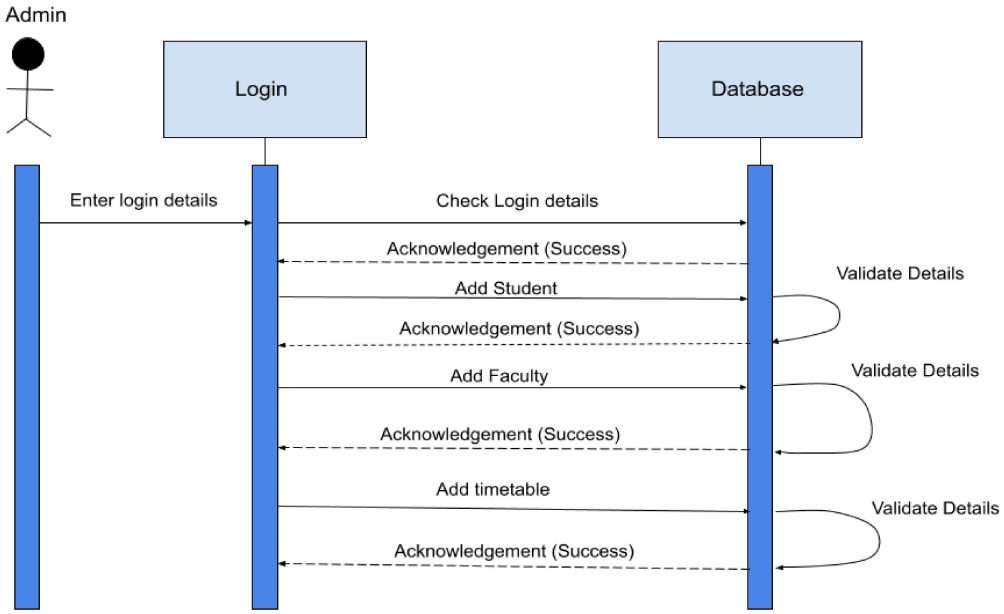

Fig. 3.5: Sequence Diagram of Admin interacting with the Automatic Attendance system

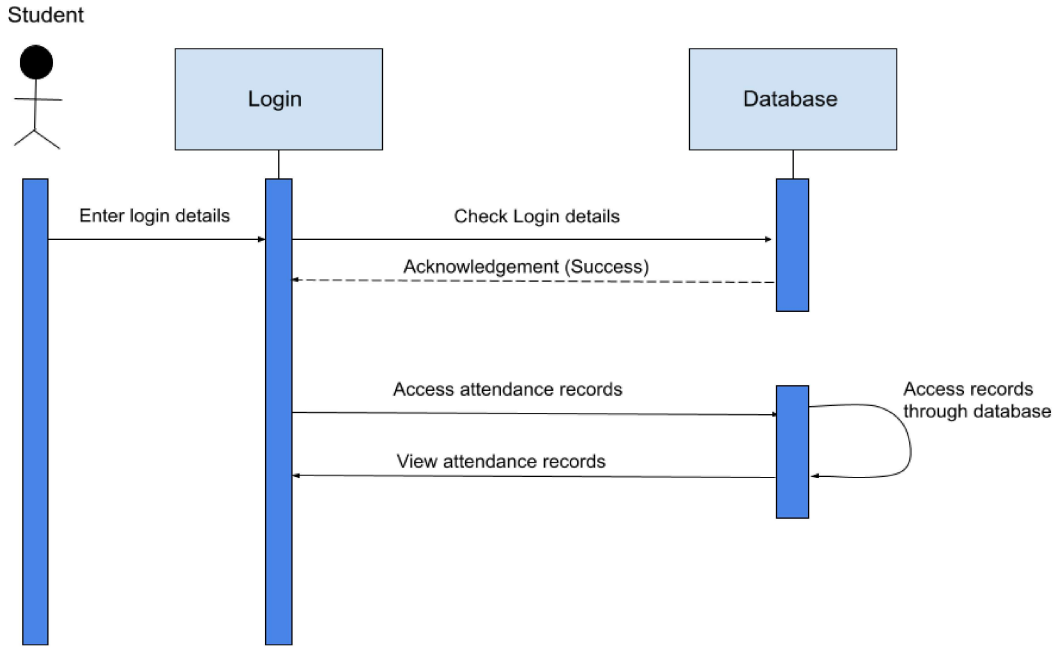

Fig. 3.6: Sequence Diagram of student interacting with Automatic Attendance system

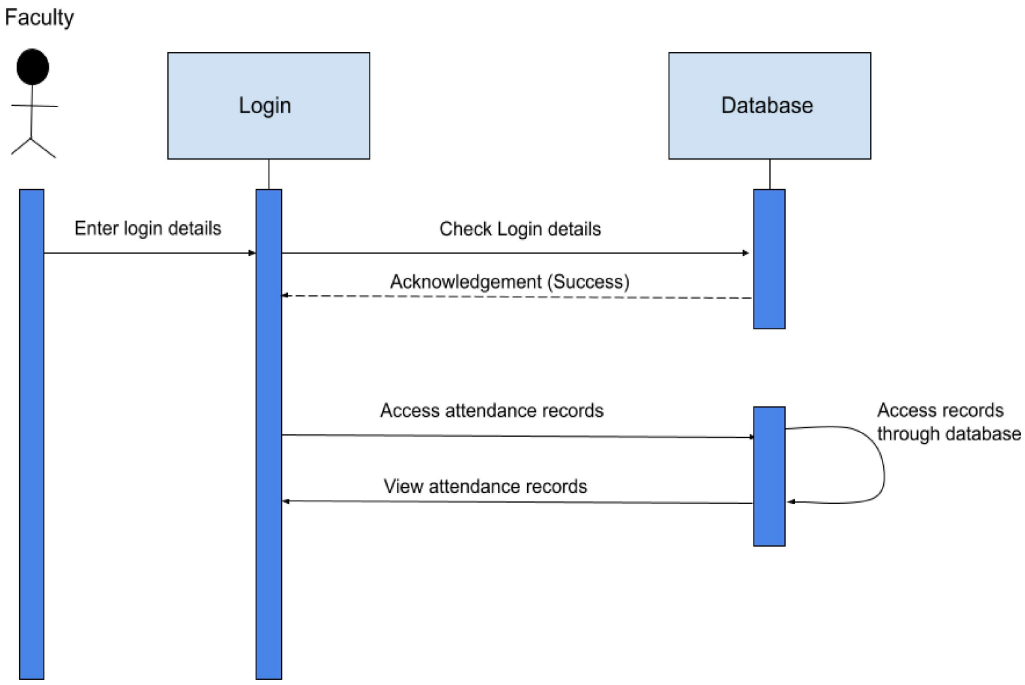

Fig. 3.7: Sequence Diagram of teacher interacting with Automatic Attendance system

The Sequence Diagram of the application consists of all the various aspects a normal sequence diagram requires. This sequence diagram shows how from the start the model flows from one step to another. Here is the sequence of all the entities linked to each other

where the user gets started with the system. In the first diagram, the user enters the system environment, the system attempts to detect the face of the user by applying the HOG algorithm, and the image operations, as well as encodings of images with that of the live captured frames, are compared to recognize if the face is that of the user or not. Then the result is marked in a CSV file and then reflected in the User Interface. In the second and third diagrams the student and admin register/login through the GUI system to view and view/edit records respectively.

3.3 Control flow Design

3.3.1 Algorithm for logic implementation

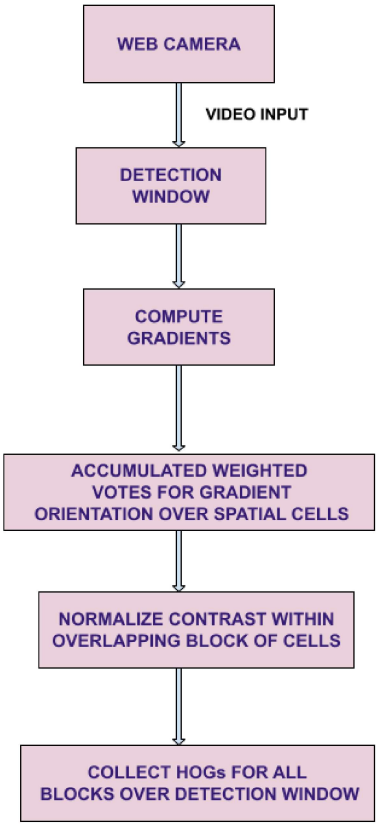

Fig. 3.8: Flowchart of HOG algorithm that is used to implement the Automated Attendance System

The above flowchart describes the HOG algorithm that we will be implementing in our project.

3.4 Access layer Design

3.4.1 Database Schema

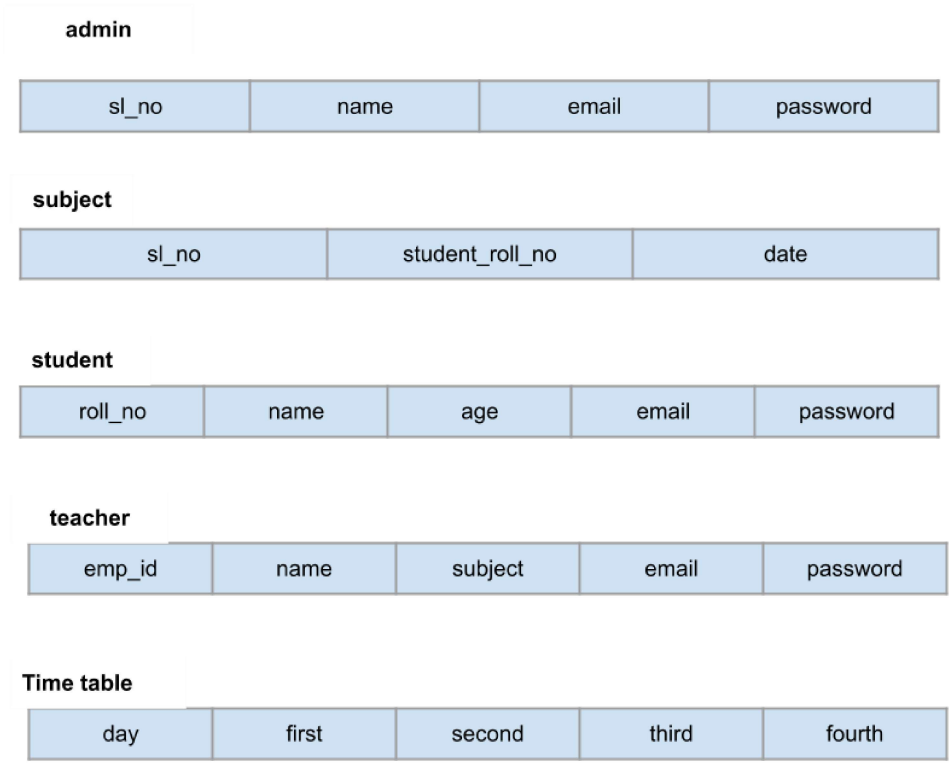

Fig. 3.9: Schema diagram of the system

The above five tables are used for our application. The admin table consists of the admin credential details. The subject table will contain the roll number of the student present on that date when it recognizes the student through the camera. The student table will consist the credentials of the students enrolled by the admin. The teacher will contain the credentials of teacher to view the attendance records that was enrolled by the admin. The timetable table specifies what subjects are present during the first four periods of each working day.

3.5 Presentation Layer Design

3.5.1 User Interface flow design

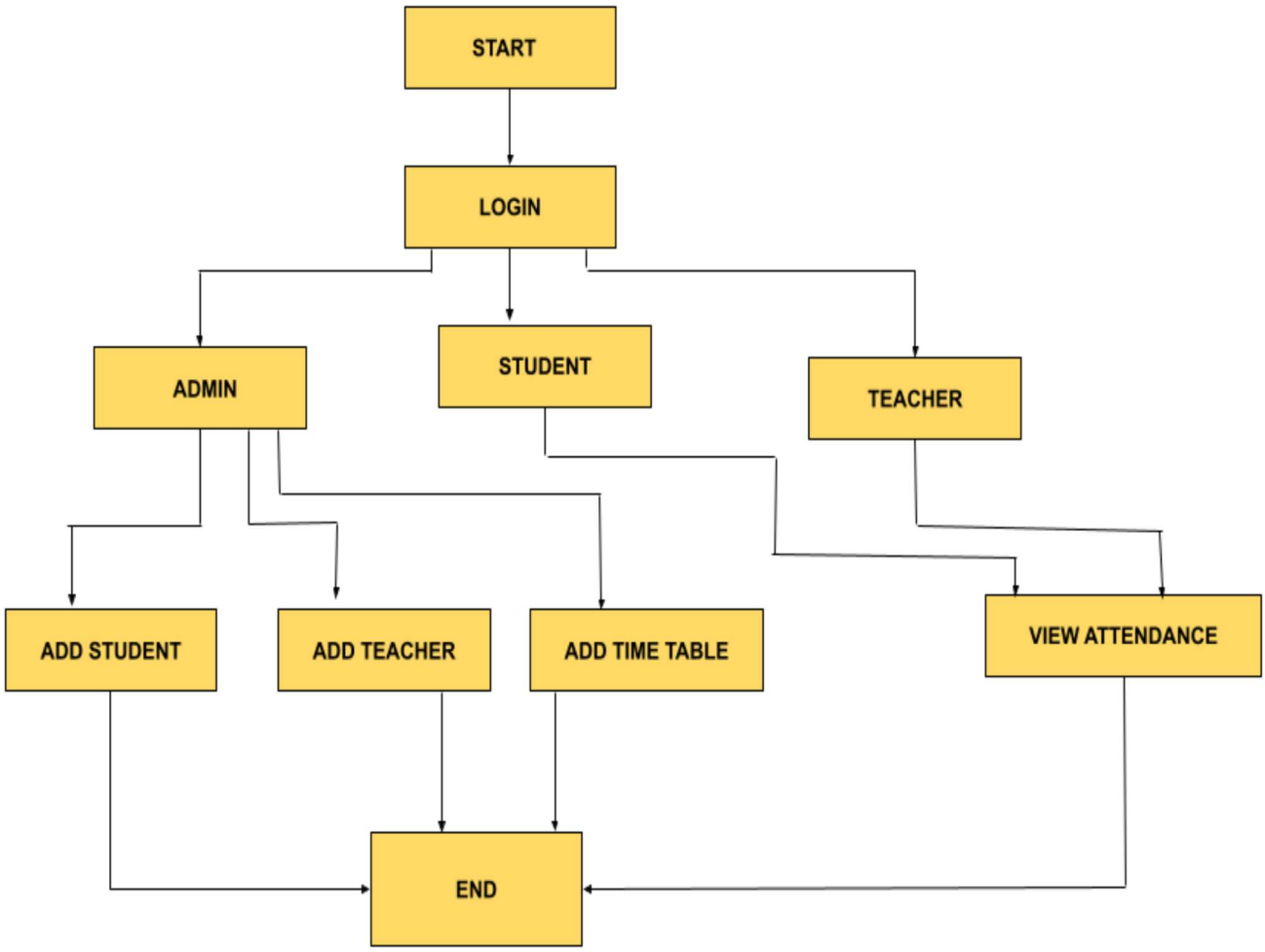

Fig. 3.10: User Interface flow diagram of Automatic Attendance system

The user interface flow diagram starts with the login page where you log in as an admin or student or as a teacher. The admin can add students, teachers, and a timetable so that students and teachers outside of our college don't try to access the system by registering them as a user where there is a high possibility of the attendance records being misused. Students can view the respective attendance records of various subjects. Whereas teachers can view the attendance records only on the subject to which they teach the students.

IMPLEMENTATION

CHAPTER 4

IMPLEMENTATION

The implementation phase is the third and most important phase of the Software Development Life Cycle (SDLC). It is the stage that converts the design into a working module. Hence it plays an important role in system development.

4.1 Software and Hardware Requirements

4.1.1 Hardware Requirements

- IP camera/web camera
- Computer
- Processor: Intel® Core™ i3-7020U CPU @2.30GHz 2.30GHz and above
- Installed RAM: 4GB (Minimum), 8GB(Recommended)

4.1.2 Software Requirements

- Windows 8 or higher
- Microsoft excel
- Python 3.5 or more
- Latest version of all libraries

Front-end: We have used HTML and CSS for the front end. HTML stands for Hyper Text Markup Language. HTML is the standard markup language for creating Web pages. HTML describes the structure of a Web page. HTML consists of a series of elements. HTML elements tell the browser how to display the content. There is no need to install any software to run HTML and CSS files. It will be opened automatically by browsers. In our project, we used HTML tags such as <form>, <input>, etc,. tags so that users can input data while using the application. We used CSS for styling purposes.

Back-end: Since our project is related to machine learning, we used python as backend programming language. Python is the best programming language for machine learning and AI. AI projects differ from traditional software projects. The differences lie in the technology stack, the skills required for an AI-based project, and the necessity of deep research. To implement your AI aspirations, you should use a programming language that is stable, flexible, and has tools available. Python offers all of this, which is why we see lots of Python AI projects today. From development to deployment and maintenance,

Python helps developers be productive and confident about the software they're building. Benefits that make Python the best fit for machine learning and AI-based projects include simplicity and consistency, access to great libraries and frameworks for AI and machine learning, flexibility, platform independence, and a wide community. These add to the overall popularity of the language.

- **Python Flask:** We created a web app using python flask framework. Flask is an API of Python that allows us to build up web-applications. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application. A Web-Application Framework or Web Framework is the collection of modules and libraries that helps the developer to write applications without writing the low-level codes such as protocols, thread management, etc. Flask is based on WSGI(Web Server Gateway Interface) toolkit and Jinja2 template engine.
- We also used python machine learning libraries such as OpenCV-python, NumPy, CMake, dlib, and face-recognition.

Tools:

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity). Visual Studio Code reduces the difficulty in programming through its enriched plugins, editor, packages and a well-built debugger. The project is developed using python, image processing packages in visual studio code, which supports image augmentation and classification.

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python programming language. It is developed by the Czech company JetBrains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control

systems (VCSes), and supports web development with Django as well as data science with Anaconda.

4.2. Pseudo Code

Pseudo code is the simplest way of representing the algorithms used in the implementation. It shows the logic of every unit of the system, hence pseudocode must be developed completely and precisely. A properly developed pseudo code helps the developers to understand the progress of their work. It also helps them not to skip any of the functionalities of the system.

The algorithm used to find and detect faces is Histogram of Oriented Gradients. Histogram of Oriented Gradients, also known as HOG, is a feature descriptor like the Canny Edge Detector, SIFT (Scale Invariant and Feature Transform) . It is used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in the localized portion of an image. This method is quite similar to Edge Orientation Histograms and Scale Invariant Feature Transformation (SIFT). The HOG descriptor focuses on the structure or the shape of an object. It is better than any edge descriptor as it uses magnitude as well as the angle of the gradient to compute the features. For the regions of the image, it generates histograms using the magnitude and orientations of the gradient.

Steps performed in the backend program used for our application:

- In the first step images of the users are stored in a local folder which are used to detect and recognize faces in real time to mark the attendance.
- In the second step, we encode the pictures using the HOG algorithm to create a simplified version of the image. Using this simplified image, find the part of the image that almost looks like a generic HOG encoding of a face.
- In the third step we figure out the pose of the face by finding the main landmarks in the face. Once we find those landmarks, use them to warp the image so that the eyes and mouth are centered.
- In the fourth step we pass the centered face image through a neural network that knows how to measure the features of the face. We then save the 128 measurements that the algorithm measures.

- In the final step, we compare faces we've measured in the past, see which person has the closest measurements to our face's measurements to find our match to the test face that was fed through the web camera, and finally the user's attendance is marked by feeding it into the database.

Working of our Web application:

- The first page appears is the index page where you can login as an admin, user or teacher. There is also a button to show the demo for the working of the backend program.
- Then clicking on any of them will redirect us to the login page.
- If logged in as an admin, he/she can add student(s), teacher(s) and timetable which can also be updated.
- If logged in as a student, he/she can view the attendance records of all the subjects.
- If logged in as a faculty, he/she can view the attendance records of the students who have the faculty's respective subject.

TESTING

CHAPTER 5

TESTING

5.1 Test Objectives

Software testing is the process of investigating, verifying and validating software or applications to ensure that they are bug free and providing the stakeholders with information related to the quality of the software or service being tested. It gives the business a wider perspective to appreciate and understand the risks involved in software implementation.

Software testing involves two steps, mainly verification and validation of properties. In general, these properties indicate the limit to which the system under test meets the requirements that lead its design and development, if the system responds accurately to vivid inputs, performs its functions within a justifiable time limit, is easily usable, can be installed and run-in intended environments, and on the whole, if it achieves the expected result its stakeholders desire.

Software testing typically aims at executing a program or application with the intention of finding software bugs and also working on improving accuracy and efficiency. Testing is carried out in iterations and is an iterative process because when one bug is resolved, it may expose another.

Some of the objectives of software testing could be summarized as follows:

- To scrutinize the errors which may get created by the programmer while developing the software.
- To gain confidence in and provide information about the level of quality.
- To avoid creating defects.
- To ensure that it satisfies the SRS that is System Requirement Specifications.
- To ensure the compatibility of the application with the OS.

5.2 Types of Testing Conducted

5.2.1 Unit Testing

In Unit testing, individual units of software are tested. Unit testing is carried out to validate each unit of the software and its performance. This type of testing is usually done by developers on the go, to ensure that each unit is working and functioning as anticipated.

Table 5.1: Use cases for unit testing

Test case ID	Test case description	Expected outcome	Observed outcome
1	Login page: when the username/password is incorrect	Displays “Invalid email or password"	Displays “Invalid email or password"
2	Login page: when the username/password is correct	Directed to the user profile	Directed to the user profile
3	In the admin profile, all entries for adding a student are not filled	Displays “All fields are necessary !”	Displays “All fields are necessary !”
4	In the admin profile, all entries for adding a student are filled	Displays “Success: Added student_name successfully”	Displays “Success: Added student_name successfully”
5	In the admin profile, while adding a student if the password and confirm password do not match	Displays “Error: password and Confirm Password doesn’t match”	Displays “Error: password and Confirm Password doesn’t match”
6	In the admin profile, while adding a student if the roll no already exists do not match	Displays “Error: Roll No exists. Provide an alternate Roll no”	Displays “Error: Roll No exists. Provide an alternate Roll no”
7	In the admin profile, all entries for adding a faculty are not filled	Displays “All fields are necessary !”	Displays “All fields are necessary !”
8	In the admin profile, all entries for adding a faculty are filled	Displays “Success: Added faculty_name successfully”	Displays “Success: Added faculty_name successfully”
9	In the admin profile, while adding a faculty if the password and confirm password do not match	Displays “Error: password and Confirm Password doesn’t match”	Displays “Error: password and Confirm Password doesn’t match”

10	In the admin profile, while adding a faculty if the employee ID already exists do not match	Displays "Error: Employee ID exists. Provide an alternate ID"	Displays "Error: Employee ID exists. Provide an alternate ID"
11	In admin profile, while not filling all the fields when adding a timetable	Displays "Error: All fields are mandatory !"	Displays "Error: All fields are mandatory !"
12	In the admin profile, while filling all the fields while adding a timetable	Displays "Table added successfully"	Displays "Table added successfully"
13	In the admin profile, while not filling all the fields when updating the timetable	Displays "Error: All fields are mandatory !"	Displays "Error: All fields are mandatory !"
14	In the admin profile, while filling all the fields when updating the timetable	Displays "Table updated successfully"	Displays "Table added successfully"
15	In the student profile, when viewing the records given the subject only and not the date	Displays the attendance records of all the dates present for that given subject	Displays the attendance records of all the dates present for that given subject
16	In the student profile, when viewing the records given the subject only and the date	Displays the attendance record of that date and if that date's record isn't found displays "Record not found.."	Displays the attendance record of that date and if that date's record isn't found displays "Record not found.."
17	In the student profile, when viewing the records by not giving details and clicking on the submit button	Displays attendance records of all the subjects present with their respective date	Displays attendance records of all the subjects present with their respective date

18	In the teacher profile, if roll no and date is not given	Displays all attendance records of the students present for that respective teacher's subject with name and date	Displays all attendance records of the students present for that respective teacher's subject with name and date
19	In the teacher profile, if roll no. and date are given	Displays the attendance record of the respective student of that date and if not present displays "Record not found…"	Displays the attendance record of the respective student of that date and if not present displays "Record not found…"
20	In the teacher profile, if roll no. is not given and date is given	Displays all the attendance records of the students present on that date else it displays "Record not found.."	Displays all the attendance records of the students present on that date else it displays "Record not found.."
21	If clicked on the logout button present in the user profile	Logs out the user and redirects to the index page	Logs out the user and redirects to the index page
22	When attendance is being taken and trying to recognize students when the backend program is running	Recognize and feed the attendance in the database	Recognizes as a student close to the dataset if the training image of that student is not present and marks the attendance of the student present among the training set even though he/she is not the student else it recognizes the student

5.2.2 Integration testing

Integration testing is where individual units are merged and tested together as a group. The aim of this level of testing is to expose incompatibilities, faults, and irregularities in the interaction between integrated modules. Integration tests involve a lot of code and may produce traces larger than unit tests.

Table 5.2: Use cases for integration testing

Test case ID	Test case description	Expected outcome	Observed outcome
1	When clicking on the button “Click on me for demo”	To run the program in the background and mark attendance live through interacting on the web interface.	The program runs in the background and marks attendance live through interacting on the web interface.
2	Email the CSV containing the attendance records to the respective subject faculty’s email	CSV is emailed to the respective subject faculty’s email ID	CSV is emailed to the respective subject faculty’s email ID

CONCLUSIONS AND FUTURE WORK

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In our project, we implemented an automatic attendance system which can save a lot of time, and effort so that faculty can avoid wasting much of their time performing a roll call. An automatic attendance system can be reliable and efficient by avoiding proxy attendance at certain times. Implementation of the system can also be useful to both students and teachers as they can simply view the attendance records through the web interface by fetching the records from the database. The system recognizes the faces of the students through which they appear within the camera's viewing window and marks attendance accordingly by having the face encoding of the training images and then compares the live image frame captured with the training image hence, attendance is marked.

In the future, the system's accuracy can be enhanced by increasing the training dataset, resolving the closed identity problem, finding an optimal solution to resolve occlusion, and facilitating the teacher to edit the records.

BIBLIOGRAPGHY

[1] https://www.ijrte.org/wp-content/uploads/papers/v9i1/A2927059120.pdf

[2] https://www.ijedr.org/papers/IJEDR1903093.pdf

[3] https://openaccess.thecvf.com/content_iccv_2015/papers/Yang_From_Facial_Parts_ICCV_2015_paper.pdf

[4] https://www.pmu.edu.sa/attachments/academics/pdf/udp/coe/dept/ee/face_detection_system_report.pdf

[5] https://www.rcciit.org/students_projects/projects/cse/2018/GR15.pdf

[6] https://www.theseus.fi/bitstream/handle/10024/503517/Facial_Recognition_Student_Attendance_System.pdf?sequence=2&isAllowe=y

[7] https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

[8] https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f#:~:text=Histogram%20of%20Oriented%20Gradients%2C%20also,the%20purpose%20of%20object%20detection.

[9] https://www.youtube.com/watch?v=sz25xxF_AVE

[10] https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world

[11] https://www.w3schools.com/

APPENDIX-1

APPENDIX-1

These are some of the screenshots of our application.

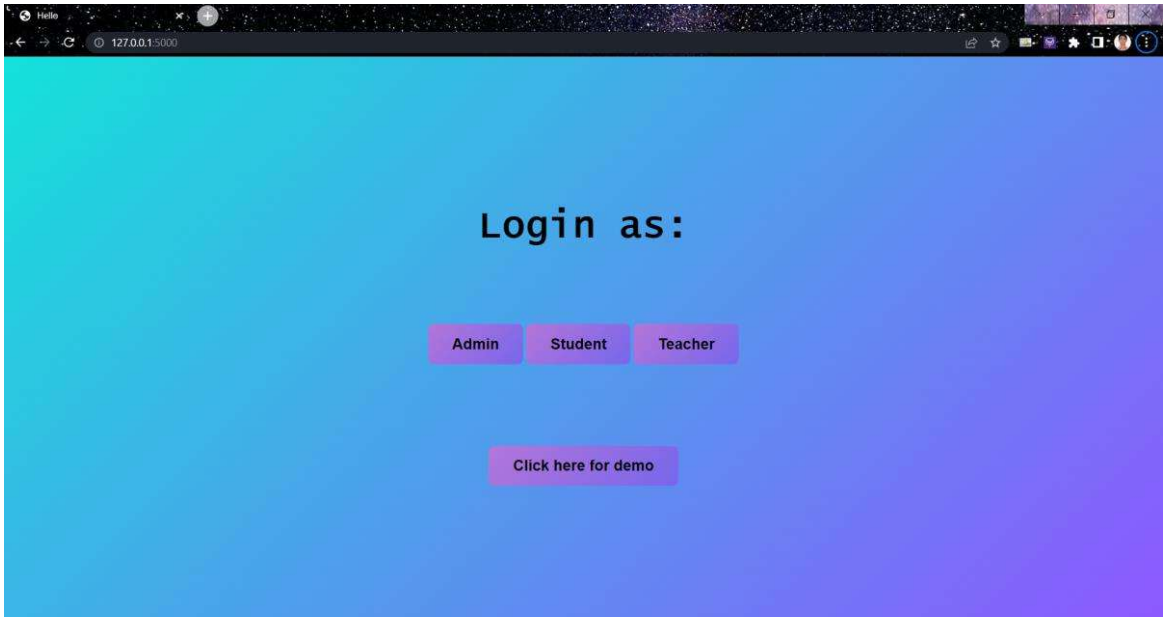

Fig. 1: Home page of application

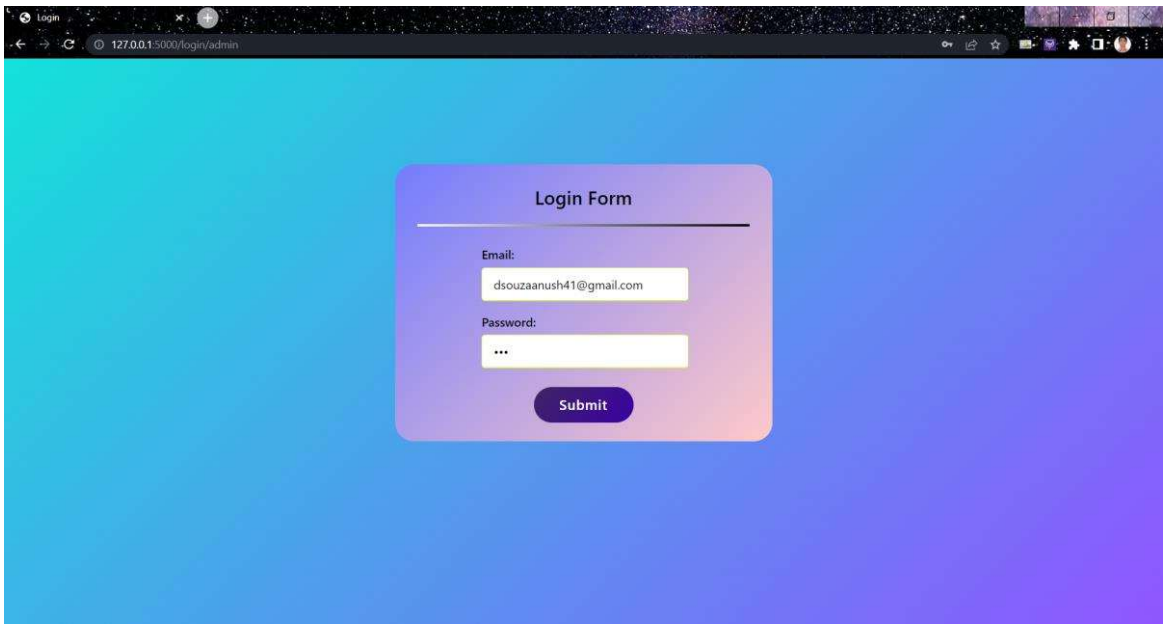

Fig. 2: Login page of application

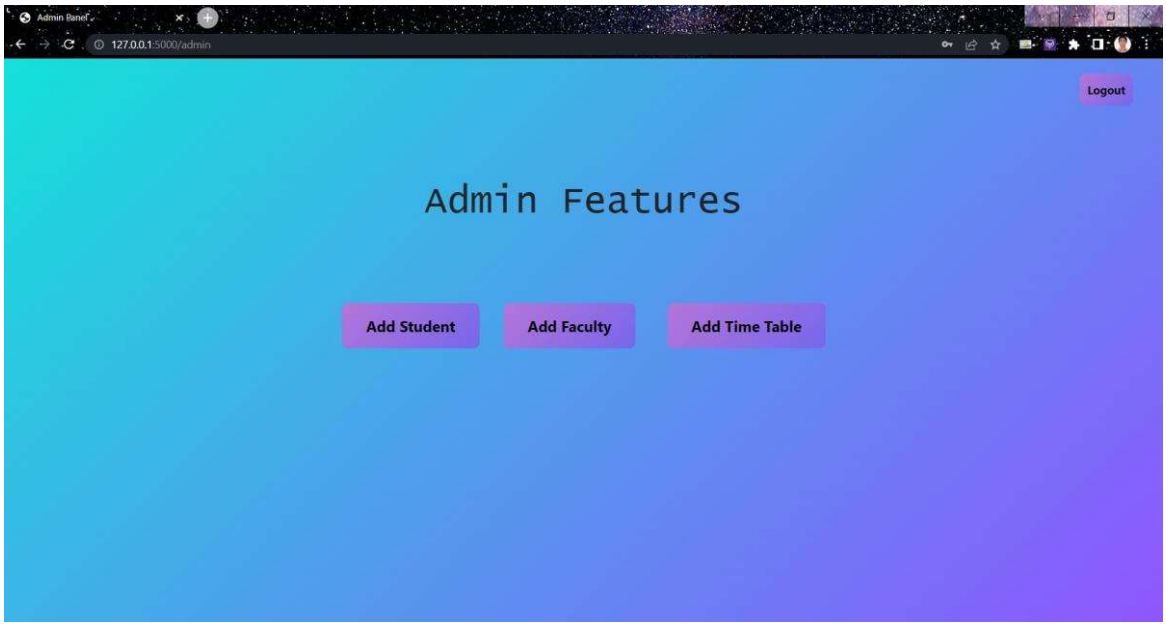

Fig. 3: Admin Panel page of application

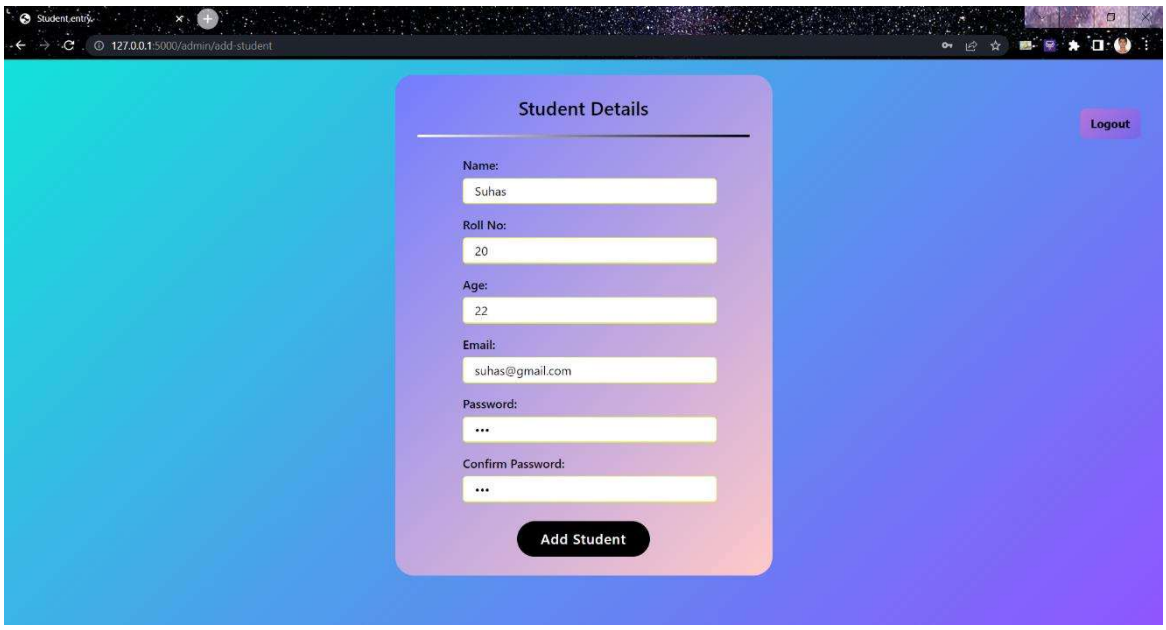

Fig. 4: Fields to enter to add a student

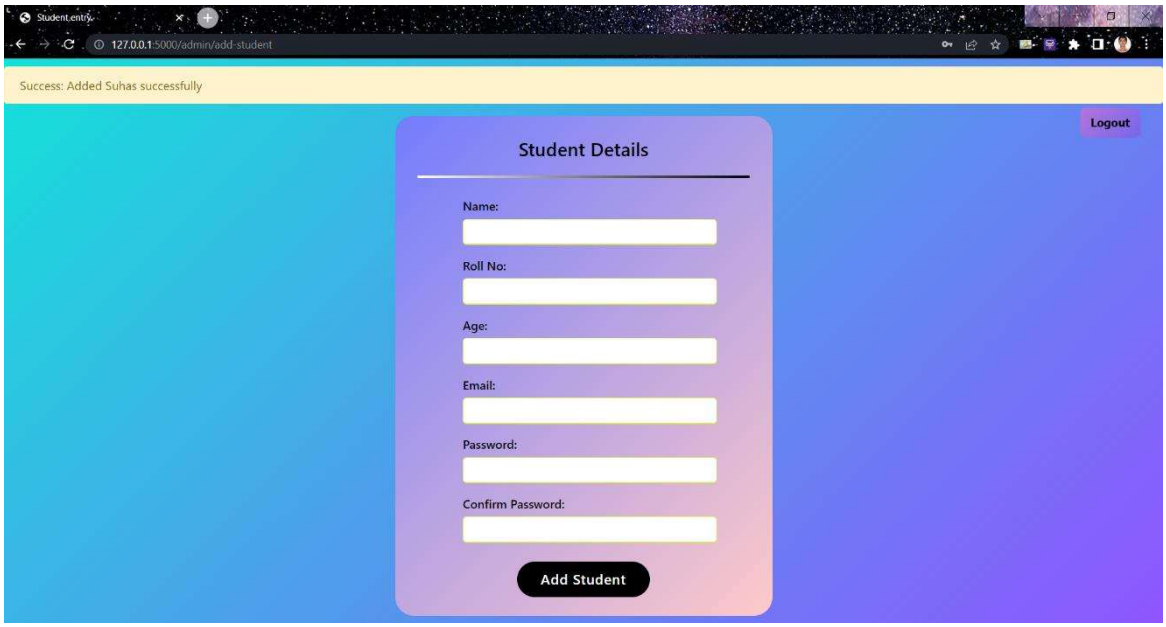

Fig. 5: Page result after adding a student

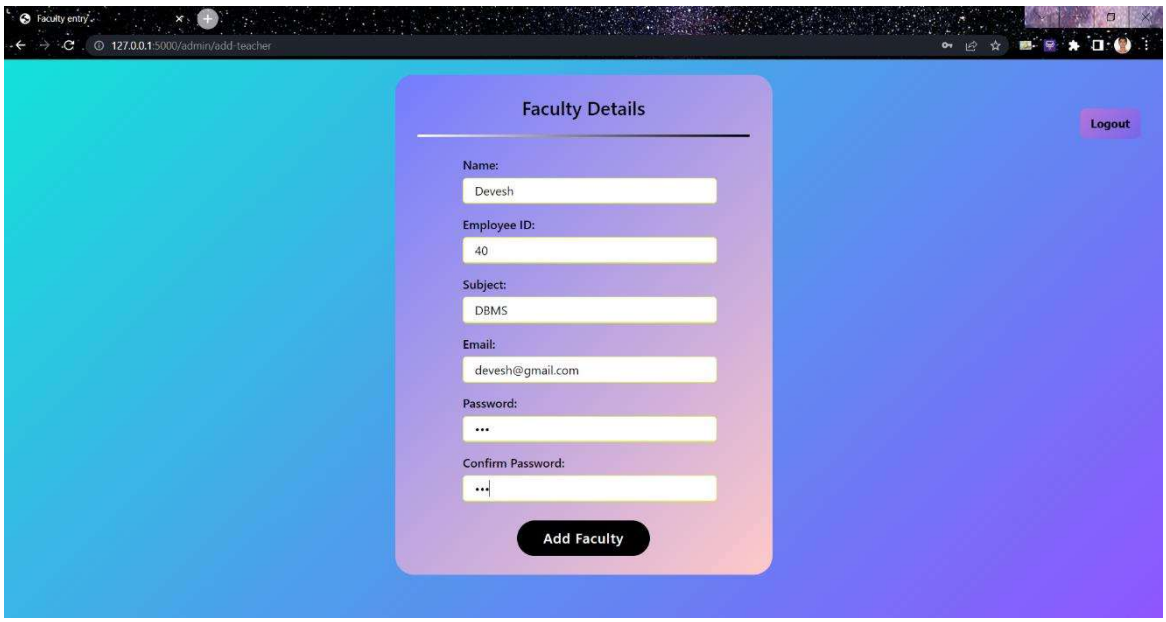

Fig. 6: Fields to be entered to add a faculty

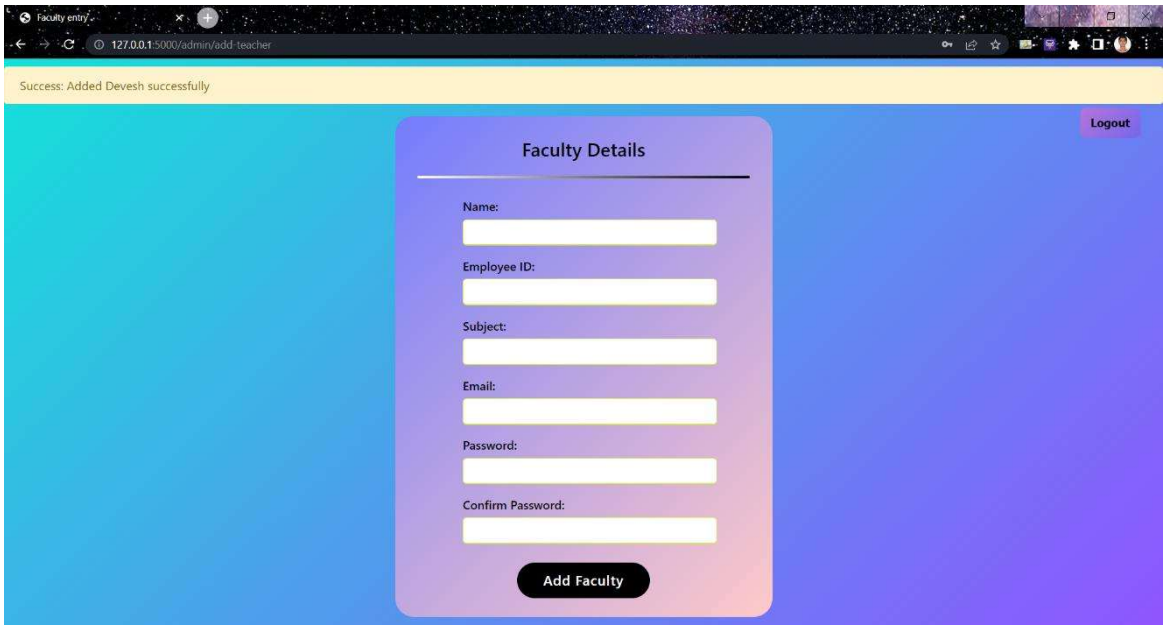

Fig. 7: Page result after adding a faculty

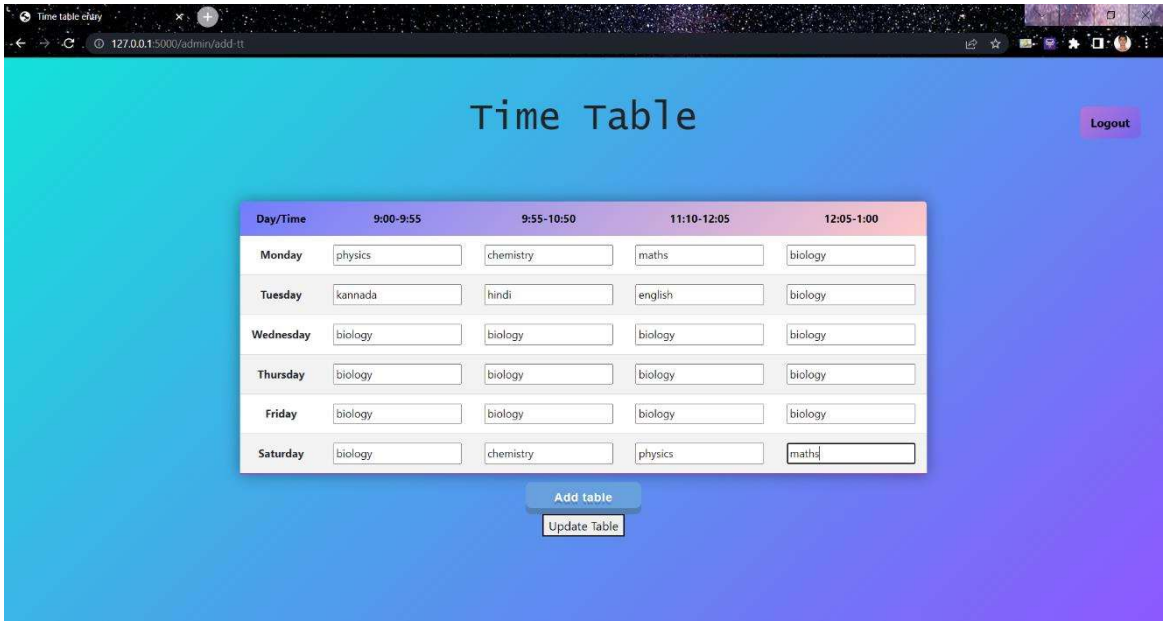

Fig. 8: Add time table page of application

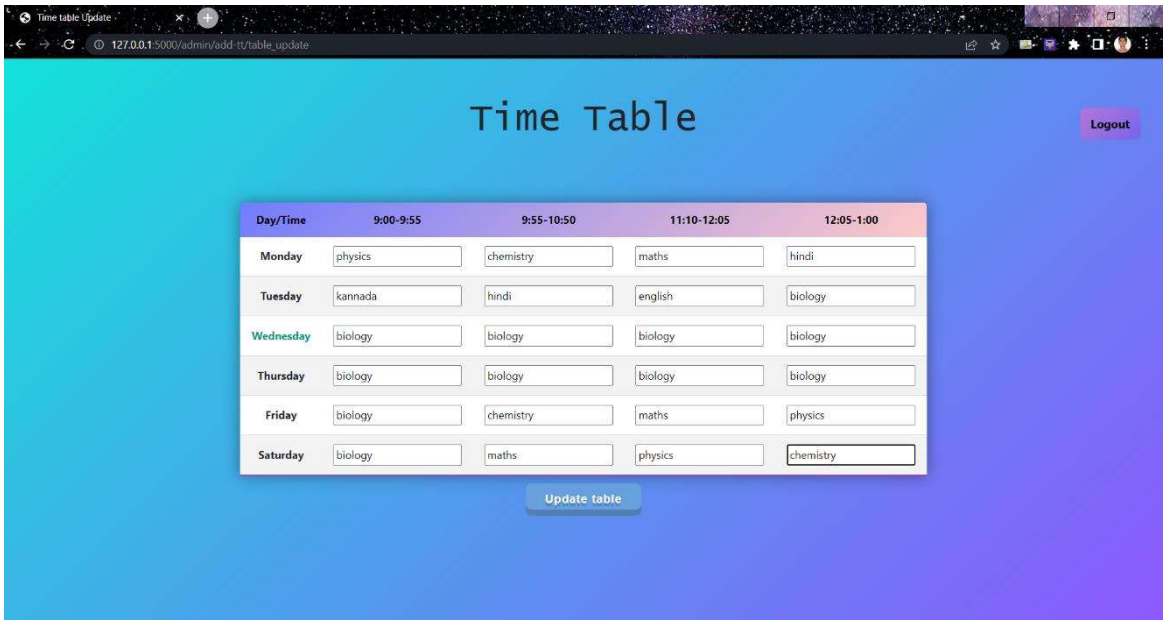

Fig. 9: Fields to be entered to insert a timetable

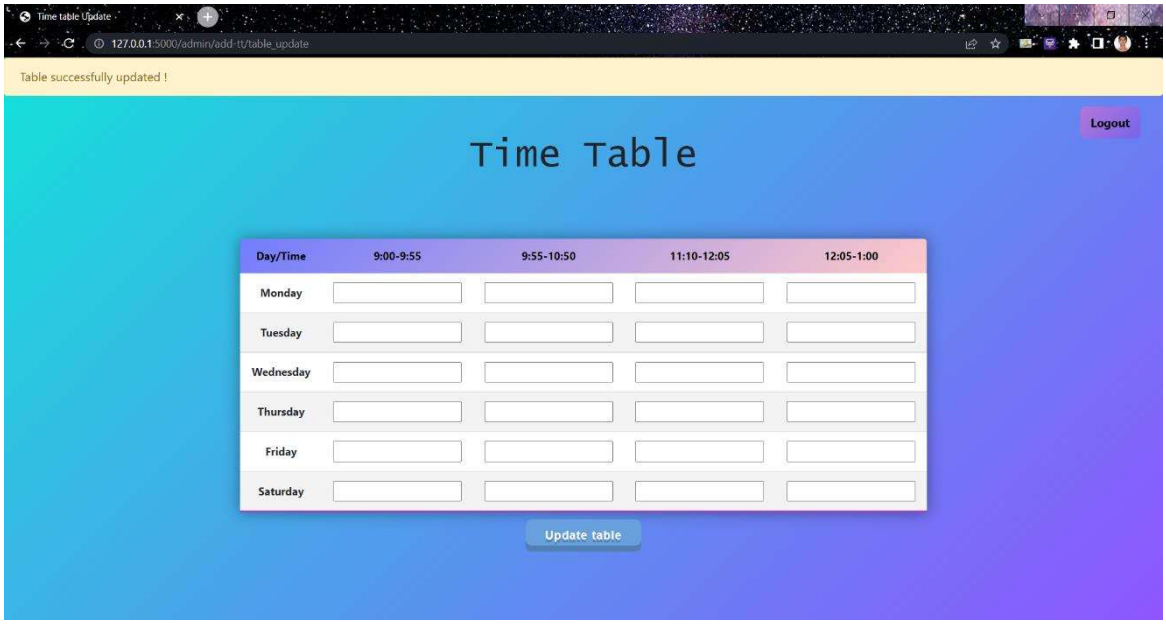

Fig. 10: Result of the page after updating a timetable

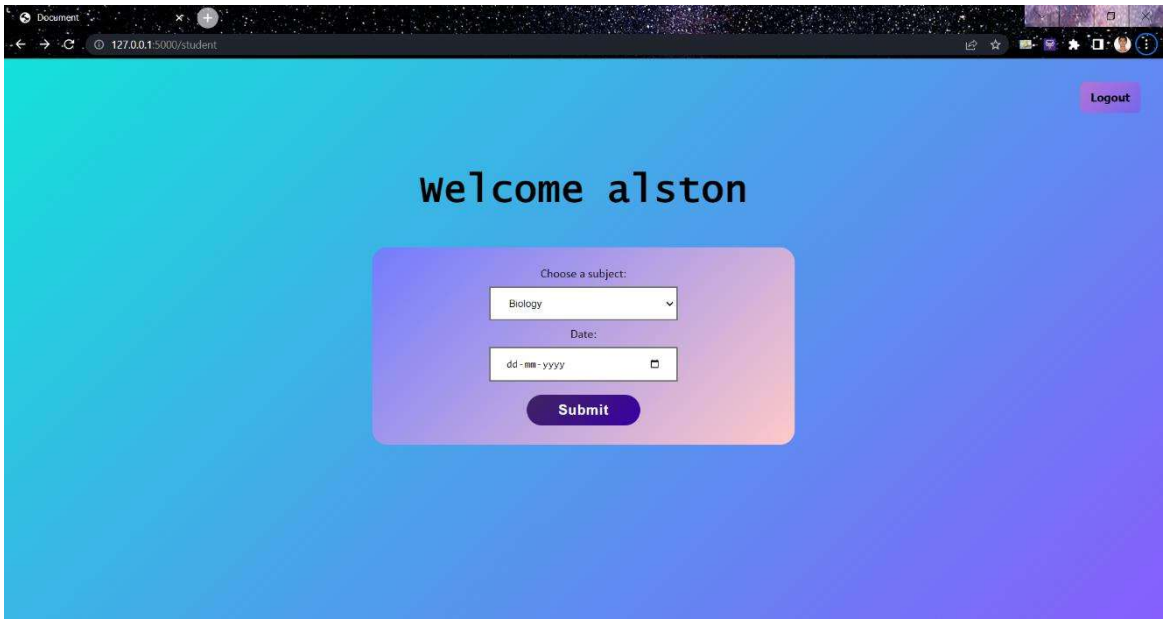

Fig. 11: Student panel page of application

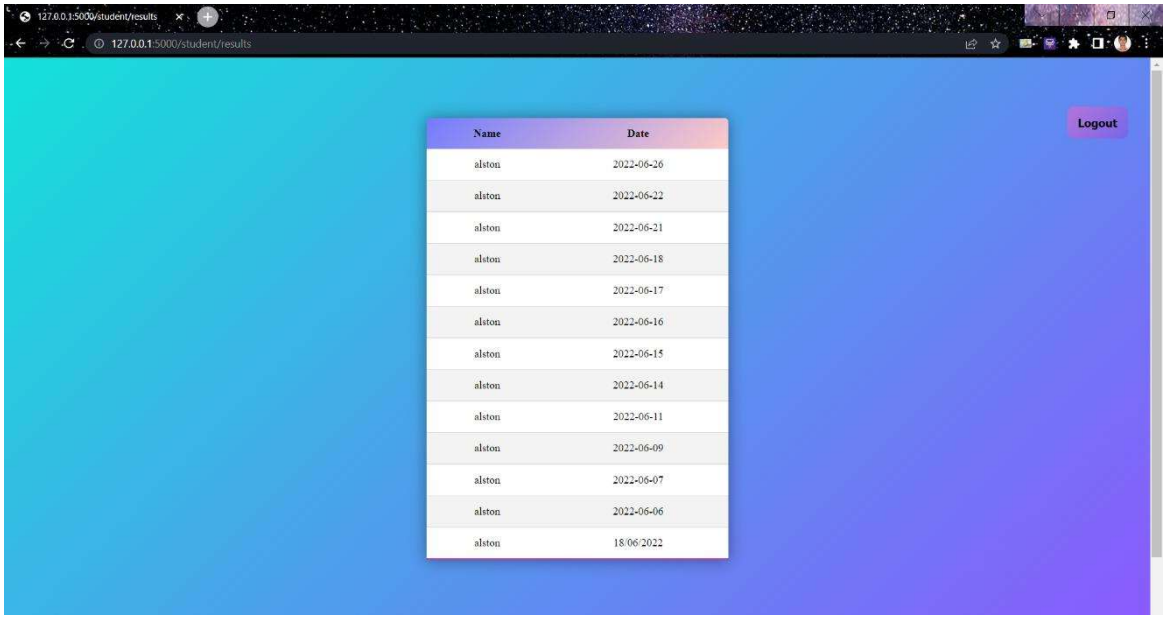

Fig. 12: Page to view attendance records

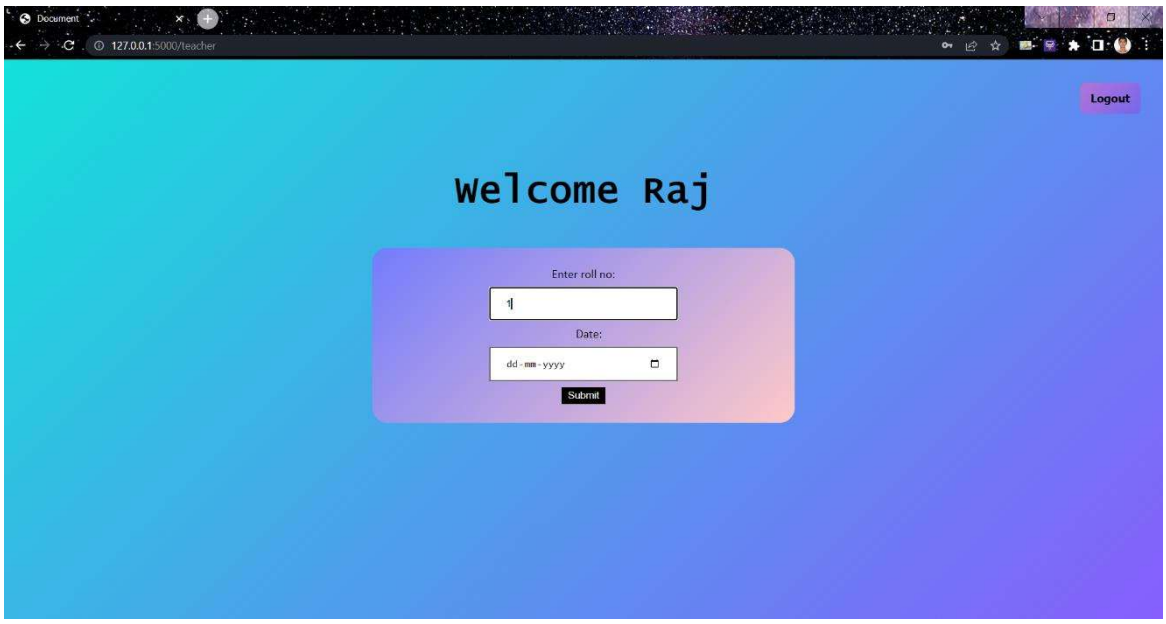

Fig. 13: Faculty panel page of application

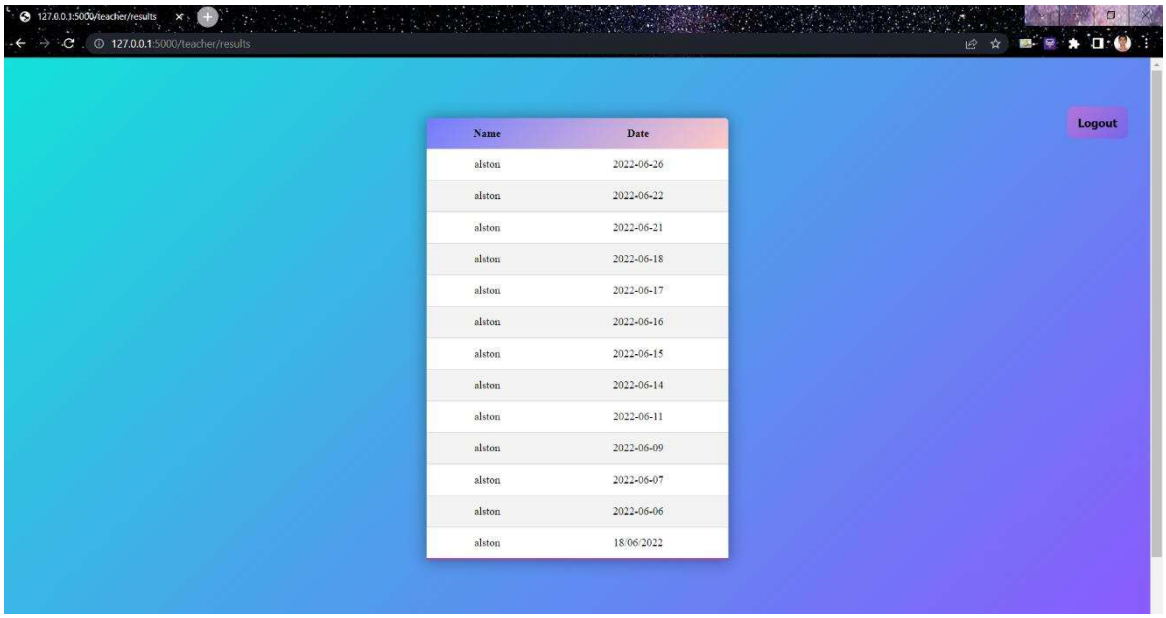

Fig. 14: Page redirected to attendance records from teacher panel

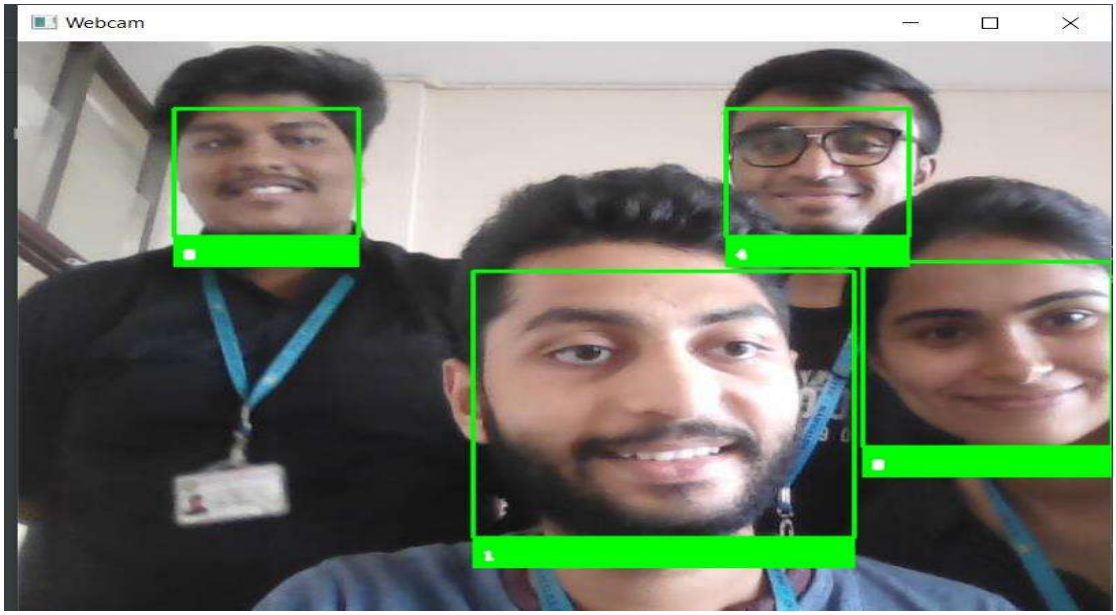

Fig. 15: Recognizing faces after running the backend program of the application