EFFICHECK

A
Mini Project Report

Submitted in partial fulfillment of
the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering

Submitted by

P. Sai Sreeya	**20SS1A0537**
K. Seemanth Raju	**20SS1A0529**
P. Prabhas Teja	**20SS1A0539**
K.J.P. Vaibhav	**20SS1A0525**
Shruti Brahma	**20SS1A0548**

Under the guidance of

Dr. G. Narsimha
Professor & Principal

Department of Computer Science and Engineering

JNTUH University College of Engineering Sultanpur

Sultanpur(V),Pulkal(M),Sangareddy district,Telangana-502273

November 2023

JNTUH UNIVERSITY COLLEGE OF ENGINEERING SULTANPUR

Sultanpur(V),Pulkal(M),Sangareddy-502273 ,Telangana

Department of Computer Science and Engineering

Certificate

This is to certify that the Mini Project report work entitled **"EFFICHECK"** is a bonafide work carried out by a team consisting of **P. Sai Sreeya** bearing **Roll no.20SS1A0537**, **K. Seemanth Raju** bearing **Roll no. 20SS1A0529**, **P. Prabhas Teja** bearing **Roll no. 20SS1A0539**, **K.J.P. Vaibhav** bearing **Roll no.20SS1A0525**, **Shruti Brahma** bearing **Roll no. 20SS1A0548** in partial fulfillment of the requirements for the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING discipline to JNTUH University College of Engineering Sultanpur during the academic year 2023- 2024.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

Guide
Dr. G. Narsimha
Professor & Principal

Head of the Department
Dr. B.V. Ram Naresh Yadav
Professor & Head of the Department

EXTERNAL EXAMINER

Declaration

This is to certify that the Mini Project report work entitled **"EFFICHECK"** is a bonafide work carried out by a team consisting of **P. Sai Sreeya** bearing **Roll no.20SS1A0537**, **K. Seemanth Raju** bearing **Roll no. 20SS1A0529**, **P. Prabhas Teja** bearing **Roll no. 20SS1A0539**, **K.J.P. Vaibhav** bearing **Roll no.20SS1A0525**, **Shruti Brahma** bearing **Roll no. 20SS1A0548** in partial fulfillment of the requirements for the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING discipline to JNTUH University College of Engineering Sultanpur during the academic year 2023- 2024.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

P. Sai Sreeya	**20SS1A0537**
K. Seemanth Raju	**20SS1A0529**
P. Prabhas Teja	**20SS1A0539**
K.J.P. Vaibhav	**20SS1A0525**
Shruti Brahma	**20SS1A0548**

Acknowledgment

We wish to take this opportunity to express our deep gratitude to all those who helped us in various ways during our Mini Project report work. It is our pleasure to acknowledge the help of all those individuals who were responsible for foreseeing the successful completion of our Major Project report.

I express our sincere gratitude to my guide **Dr. G. Narsimha, Professor & Principal, JNTUHUCES** for his support during the course period.

I sincerely thank **Dr. Y. Raghavender Rao, Professor of ECE & Vice Principal, JNTUHUCES** for his kind help and co-operation.

I am thankful to **Dr. B. V. Ram Naresh Yadav, Professor & Head of the Department**, for his support and guidance during the course period .

Finally, Special thanks to my parents for their moral support and encouragement throughout this course.

P. Sai Sreeya	**20SS1A0537**
K. Seemanth Raju	**20SS1A0529**
P. Prabhas Teja	**20SS1A0539**
K.J.P. Vaibhav	**20SS1A0525**
Shruti Brahma	**20SS1A0548**

Contents

Abstract

This logbook system revolves around utilizing identity cards to manage and track student attendance efficiently. The system captures essential attributes from identity cards, including roll number and name, while also recording the time of entry and exit for each student. By integrating this data, the system can promptly identify latecomers and subsequently notify them. Additionally, students and faculty can conveniently check their own in and out times through this system. This abstract explores the design and functionality of the logbook system, highlighting its potential to streamline attendance monitoring and enhance punctuality among students. The innovative utilization of identity cards, coupled with automated notifications, presents a comprehensive solution to address attendance-related challenges in educational institutions.

List of Figures

Chapter 1

INTRODUCTION

1.1 Project Overview

The innovative logbook system seamlessly tracks student attendance by enabling efficient in and out time management. The logbook system efficiently tracks student attendance by utilizing identity cards containing essential information like roll numbers and names for streamlined in and out tracking. The logbook system tracks critical attendance data by recording the time of entry and exit for each student. The logbook system captures essential data including roll numbers, names, entry times and exit times by using student identity cards to register in and out of designated institutional locations. By integrating collected entry time data, the logbook system promptly identifies late students and triggers notifications to inform them of their tardiness.

1.2 Purpose

Optimize student and faculty efficiency by simplifying essential tasks and responsibilities.

- Efficient Tracking with Identity Cards: Streamlines student attendance monitoring by capturing roll numbers, names, entry, and exit times using identity cards,

enhancing the overall process.

- Automated Latecomer Identification: Promotes punctuality by automatically identifying latecomers and notifying them, reducing administrative burden on faculty and staff.
- User-Friendly Access: Provides convenient access for students and faculty to check their attendance records, fostering transparency and accountability within the educational community.

1.3 Existing System

- Relies on manual attendance sheets and occasional barcode scanning. Limited automation, leading to inefficiencies and potential errors. Minimal real-time monitoring capabilities.
- Lacks intuitive interfaces for both students and faculty. Inconsistent feedback mechanisms hinder user experience. Limited accessibility for individuals to check their attendance records.
- Relying on basic authentication with potential vulnerabilities. Inadequate access control measures. Limited protection against potential data breaches.

1.4 Proposed System

This proposed system addresses the limitations of traditional methods by introducing modern technologies, enhancing user experiences, improving security measures, and automating the attendance tracking process.

- ID Card-Based Tracking: Introduces streamlined ID card scanning for entry and exit. Enables real-time data capture for accurate attendance tracking. Modernizes and automates the attendance process.
- Enhanced User Interfaces: Implements intuitive interfaces for a positive user experience. Provides real-time feedback on ID card scanning actions. Facilitates convenient access for users to check their attendance records.

- Robust Security Measures: Implements advanced authentication mechanisms. Integrates encryption protocols for secure data transmission. Establishes comprehensive access controls for enhanced security.

1.5 Scope

The scope of the proposed system is to revolutionize attendance tracking in educational institutions by introducing a modern ID card-based approach. This system aims to streamline the process through efficient entry and exit logging, providing real-time feedback to students and faculty. Enhanced user interfaces, robust security measures, and integration of emerging technologies contribute to an overall improved attendance management system. The scope extends to addressing existing challenges in traditional methods, offering a user-friendly experience, and fostering a secure and technologically advanced environment for attendance monitoring, ultimately contributing to a more efficient and accountable educational ecosystem.

Chapter 2

LITERATURE SURVEY

The proposed online ebook model with ID card-based attendance tracking draws upon a rich body of existing research and technologies related to similar systems. Several key themes emerge from the literature, contributing valuable insights and shaping the foundation for the proposed system.

Attendance Tracking Systems
Existing studies highlight various methodologies for attendance tracking in educational settings. Krishna and Reddy (2014) propose a novel approach using barcode technology, while Patil, Patil, and Deogirikar (2017) explore QR code-based attendance systems. These methodologies inform the design considerations for the proposed ID card-based system.

User Interface and Experience
Literature emphasizes the significance of user interfaces in educational technology. Al-Hamad and Huda (2016) investigate a face recognition-based attendance system for classroom monitoring, while Rahman, Hossain, and Rahman (2016) focus on an RFID-based attendance management system. These works underscore the importance of intuitive interfaces for ID card scanning, real-time feedback, and user-friendly dashboards for attendance monitoring.

Security Measures
Security considerations are crucial in attendance tracking systems. Dwivedi and Misra (2015) propose a secure cloud-based attendance system using fingerprint and RFID technology, while Patil and Patil (2016) focus on an attendance management system

using RFID and GSM technology. Understanding these security aspects informed the integration of a robust security module in the proposed architecture.

Integration of Notifications

Literature reveals that real-time notifications play a pivotal role in engaging users and ensuring timely actions. Al-Deek and Al-Jabri (2015) explore a smart attendance system using RFID and SMS technology, while Rahman, Rahman, and Islam (2016) propose a cloud-based attendance management system with RFID and GSM technology. This insight informed the optional inclusion of a notification interface in the proposed system.

Database Management

Effective database management is crucial for maintaining accurate attendance records. Patil and Patil (2016) focus on designing and implementing an attendance management system using RFID and GSM technology, while Dwivedi and Misra (2015) emphasize the role of databases in storing and retrieving information efficiently. This knowledge contributed to the design of a robust database layer in the proposed architecture.

Technological Trends

Emerging technological trends, such as machine learning and data analytics, are increasingly being applied in educational systems. Patil, Patil, and Deogirikar (2017) highlight the potential of QR code-based attendance systems, while Krishna and Reddy (2014) propose a novel approach using barcode technology. The literature survey reveals potential avenues for future enhancements, including predictive analytics for attendance patterns and the incorporation of AI-driven features.

Challenges and Solutions

Existing literature sheds light on challenges faced by attendance tracking systems, including issues related to accuracy, user compliance, and system reliability. Al-Hamad and Huda (2016) discuss the accuracy of face recognition-based systems, while Rahman, Hossain, and Rahman (2016) emphasize the importance of user compliance in RFID-based systems. Understanding these challenges informed strategies for mitigating potential issues in the proposed model.

In conclusion, the literature survey provides a comprehensive overview of existing research and technologies related to ID card-based attendance tracking systems. The insights gained from this survey informed the design and development of the proposed online ebook model.

Chapter 3

REQUIREMENT SPECIFICATION

3.1 Software Requirements

- IDE Anakonda/Google Collab
- Python3.6 or higher, React
- Unix Based / Windows 8 or Higher
- Latest version of all python libraries viz Django, Easyocr.
- Latest version of any web browser.

3.2 Hardware Requirements

- Minimum Processor Intel i5
- Minimum 2 GHz CPU
- Minimum 4GB RAM
- Minimum hard disk 80GB
- Input Devices viz. Camera, Keyboard, and Mouse

3.3 Conclusion

In conclusion, the outlined software requirements, encompassing Anaconda or Visual Studio Code, Python 3.6+, and compatibility with Unix and Windows 8+, ensure a versatile and up-to-date development environment. Emphasizing the latest versions of Django and Easyocr libraries promotes feature-rich and secure application development. Additionally, the insistence on the latest web browser aligns with modern web standards. On the hardware front, the project demands a minimum Intel i5 processor, 4GB RAM, and an 80GB hard disk, guaranteeing efficient performance and scalability. The inclusion of essential input devices such as a camera, keyboard, and mouse signifies comprehensive user interaction capabilities. This well-documented set of requirements establishes a robust foundation for successful project development and deployment.

Chapter 4

WORKING OF SYSTEM

4.1 SYSTEM ARCHITECTURE

The system architecture for the online ebook model, featuring ID card-based attendance tracking, is designed to seamlessly integrate frontend and backend components for optimal functionality. The frontend encompasses user interfaces tailored for students, faculty, and administrators, ensuring a user-friendly experience. Students can efficiently scan their ID cards, receive real-time feedback, and access personal attendance records. Faculty members benefit from a dedicated interface for viewing attendance records, configuring system settings, and generating reports. The centralized Admin Panel empowers administrators with tools to configure system settings and manage user accounts.

On the backend, the architecture includes modules for ID card scanning, data processing, and a robust database layer for storing student information and attendance records. An optional notification service can be integrated to send real-time alerts based on system events. Additionally, a security module is implemented to enforce access controls, ensuring data integrity and user authentication. This cohesive system architecture supports efficient attendance tracking, user management, and data security in the dynamic context of an online ebook model.

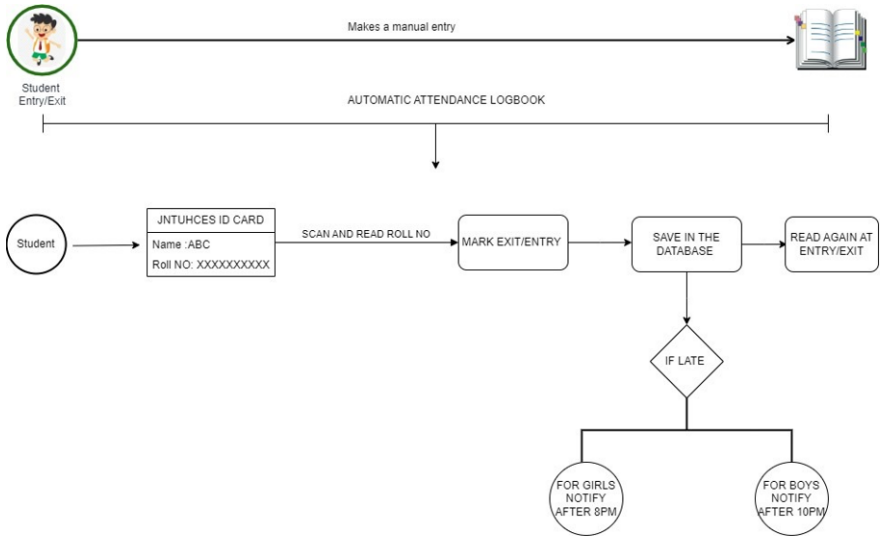

Figure 4.1: System Architecture

4.2 MACHINE LEARNING

4.2.1 Natural Language Processing (NLP) Modules

User Interface Module: Provides a user-friendly interface for student guard, and administrators to interact with the system.

Identity Card Reader Module: Responsible for capturing essential attributes from student identity cards. Extracts information such as roll number and name from the identity cards.

Time Tracking Module: Records the entry and exit times for each student. Captures the timestamp when a student presents their identity card. Algorithmic analysis of entry times, potentially leveraging timestamp data.

4.2.2 Back-End Service Modules:

Utilizes a relational database system i.e Django for storing student identity information, attendance records, and relevant data. Stores the authorization data of administrator and

security guard.

API Integration: This module integrates with third-party APIs to provide additional functionality to the virtual assistant. For example, the virtual assistant can use Google Maps APIs to provide real-time navigation directions.

User Authentication: This module handles user authentication and authorization. This is important for ensuring that only authorized users can access the system.

In addition to the specific modules listed above, machine learning can also be used in other parts of the system, such as:
Personalization: Machine learning can be used to personalize the user experience by recommending activities, events, and resources that are relevant to the user's interests.

Navigation: Machine learning can be used to improve the accuracy of navigation directions by taking into account factors such as traffic conditions and weather.

Chapter 5

SYSTEM DESIGN

Design is the abstraction of a solution it is a general description of the solution to a problem without the details. Design is view patterns seen in the analysis phase to be a pattern in a design phase. After design phase we can reduce the time required to create the implementation.

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

What is UML?
UML is an acronym that stands for Unified Modelling Language. Simply put, UML is[7] a modern approach to modelling and documenting software. In fact, it's one of the most popular business process modelling techniques.
It is based on diagrammatic representations of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

Building Blocks of the UML: The vocabulary of the UML encompasses three kinds of building blocks.

- **Things:** Things are the abstractions that are first-class citizens in a model
- **Relationships:** ; relationships tie these things together

- **Diagrams:** diagrams group interesting collections of things

5.1 Use Case Diagram

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system.

A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors. In this project, JNTUHUCES Student, Guest/new Admission, Admin are the actors

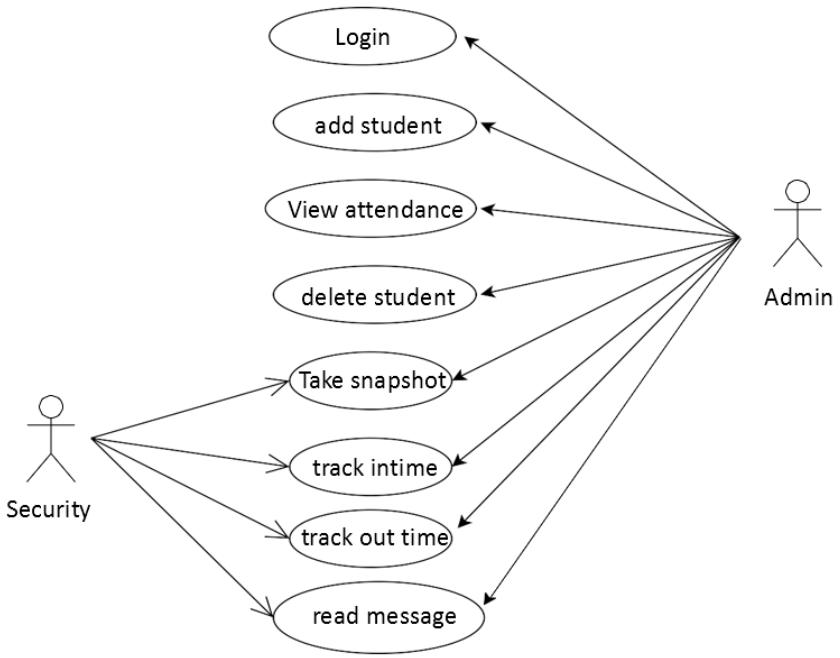

Figure 5.1: Use Case diagram

5.2 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

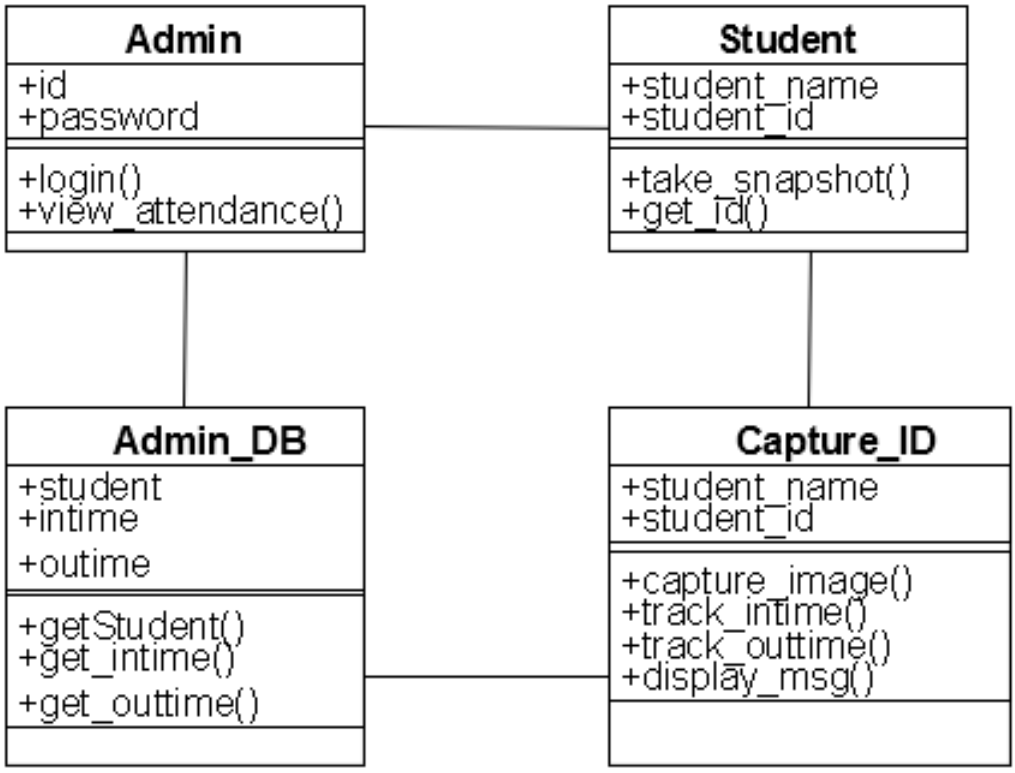

Figure 5.2: Class diagram

5.3 Sequence Diagram

A sequence diagram is a visual representation that illustrates the interactions between different components or objects in a system over time. It shows the flow of messages or events between these components, providing a dynamic view of system behavior.

In the context of an attendance e-book system, a sequence diagram would depict the chronological order of interactions between various elements involved in recording and managing attendance.

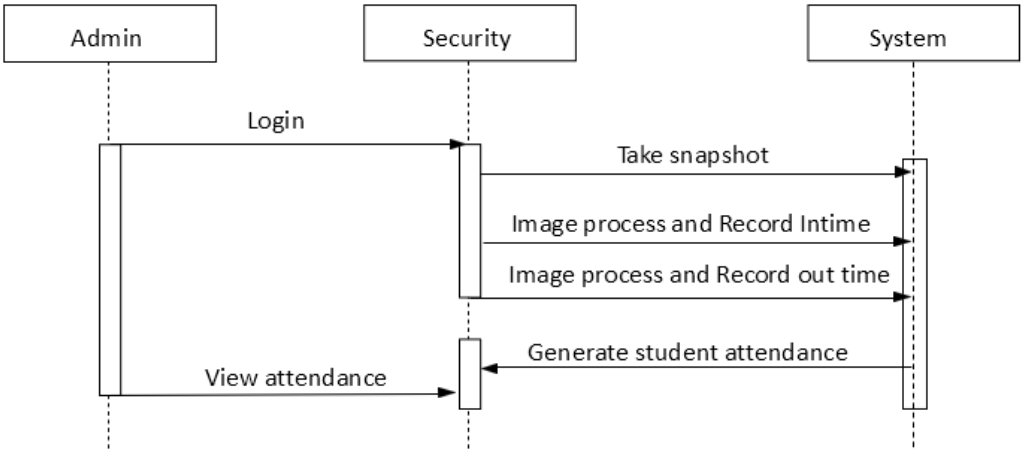

Figure 5.3: Sequence Diagram

5.4 Flow Chart Diagram

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

The flowchart shows[3] the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

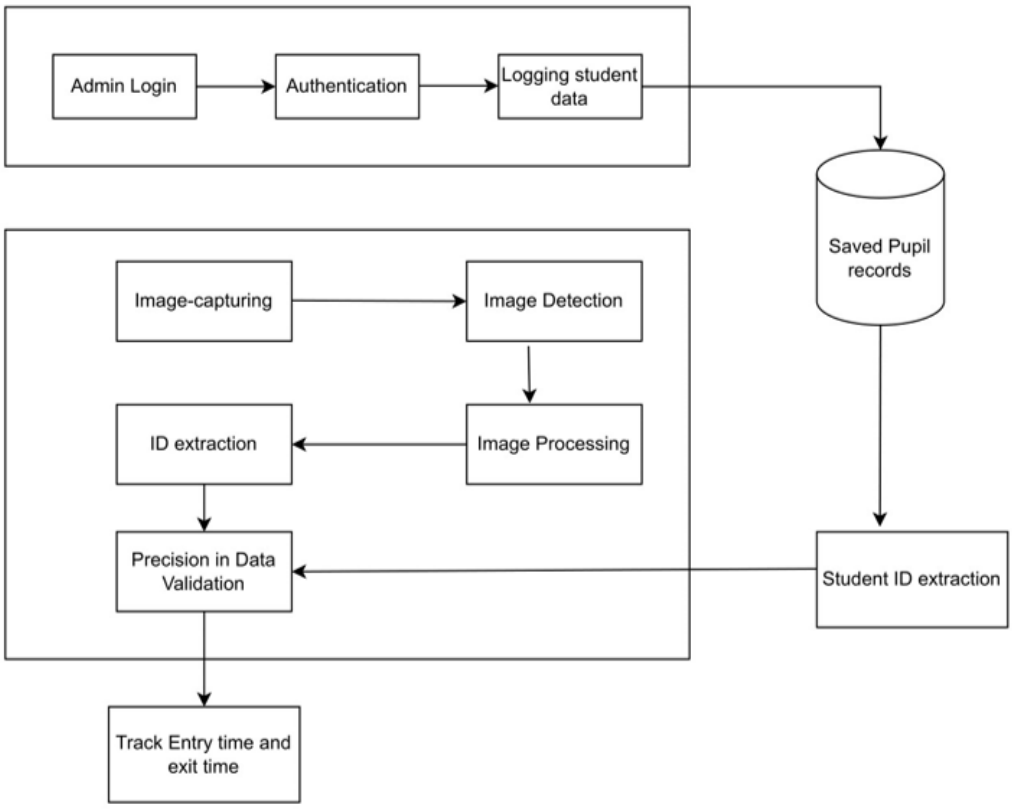

Figure 5.4: Flow Chart diagram

5.5 Collaboration Chart Diagram

A collaboration diagram, also known as a communication diagram, illustrates how different components or objects in a system interact with each other to achieve a specific goal. In the context of a collaboration diagram for an attendance e-book system, we can visualize the communication between system entities.

This collaboration diagram provides a dynamic representation of how different components collaborate to manage attendance efficiently. It emphasizes the flow of communication between student entities, the attendance system, and associated modules, offering insights into the collaborative nature of the attendance e-book system..

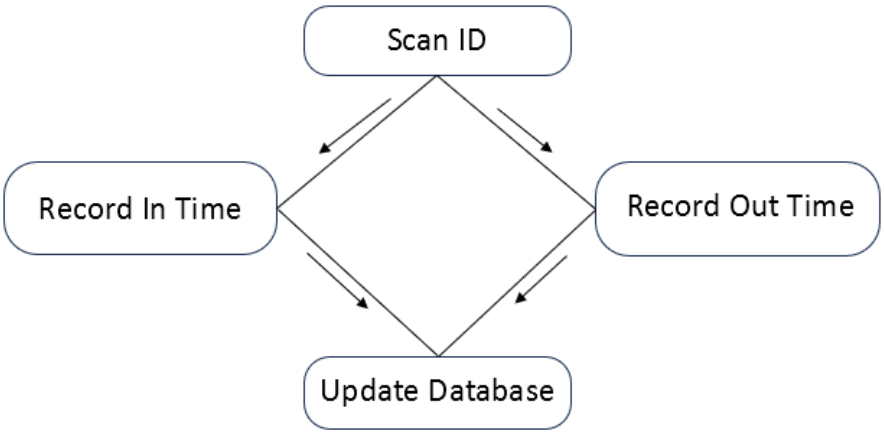

Figure 5.5: Collaboration diagram

Chapter 6

IMPLEMENTATION

6.1 Directory

```
Efficheck
├── db
├── manage.py
├── Project
│   ├── __pycache__
│   ├── __init__.py
│   ├── asgi.py
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
└── App
    ├── __pycache__
    ├── __init__.py
    ├── admin.py
    ├── apps.py
    ├── models.py
    ├── tests.py
    ├── views.py
    └── templates
        ├── login.html
        ├── home.html
        ├── base.html
        ├── login.html
        ├── studentReg.html
        └── studentDataFetch.html
```

6.2 Efficheck

manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys


def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', '
    StudentEntryExit.settings')
    try:
        from django.core.management import
    execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed
    and "
            "available on your PYTHONPATH environment variable?
    Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)


if __name__ == '__main__':
    main()
```

Listing 6.1: *manage.py*

6.2.1 Project Folder

settings.py

```
"""
Django settings for StudentEntryExit project.

Generated by 'django-admin startproject' using Django 4.1.7.

"""
```

```
7
8  from pathlib import Path
9  import os
10
11 # Build paths inside the project like this: BASE_DIR / 'subdir'.
12 BASE_DIR = Path(__file__).resolve().parent.parent
13
14
15
16 # SECURITY WARNING: keep the secret key used in production secret!
17 SECRET_KEY = 'django-insecure-s$bfl!o4iqpt-7@a)$s)#pl9mea4%8
       looglzo=y=cXXXXXXXX'
18
19 # SECURITY WARNING: don't run with debug turned on in production!
20 DEBUG = True
21
22 ALLOWED_HOSTS = []
23
24
25 # Application definition
26
27 INSTALLED_APPS = [
28     'django.contrib.admin',
29     'django.contrib.auth',
30     'django.contrib.contenttypes',
31     'django.contrib.sessions',
32     'django.contrib.messages',
33     'django.contrib.staticfiles',
34     'pro'
35 ]
36
37 MIDDLEWARE = [
38     'django.middleware.security.SecurityMiddleware',
39     'django.contrib.sessions.middleware.SessionMiddleware',
40     'django.middleware.common.CommonMiddleware',
41     'django.middleware.csrf.CsrfViewMiddleware',
42     'django.contrib.auth.middleware.AuthenticationMiddleware',
43     'django.contrib.messages.middleware.MessageMiddleware',
44     'django.middleware.clickjacking.XFrameOptionsMiddleware',
45 ]
46
47 ROOT_URLCONF = 'StudentEntryExit.urls'
48
49 TEMPLATES = [
50     {
51         'BACKEND': 'django.template.backends.django.
```

```
    DjangoTemplates',
52          'DIRS': [os.path.join(BASE_DIR,"templates")],
53          'APP_DIRS': True,
54          'OPTIONS': {
55              'context_processors': [
56                  'django.template.context_processors.debug',
57                  'django.template.context_processors.request',
58                  'django.contrib.auth.context_processors.auth',
59                  'django.contrib.messages.context_processors.
    messages',
60              ],
61          },
62      },
63 ]
64
65 WSGI_APPLICATION = 'StudentEntryExit.wsgi.application'
66
67
68 # Database
69
70 DATABASES = {
71      'default': {
72          'ENGINE': 'django.db.backends.sqlite3',
73          'NAME': BASE_DIR / 'db.sqlite3',
74      }
75 }
76
77
78 # Password validation
79
80 AUTH_PASSWORD_VALIDATORS = [
81      {
82          'NAME': 'django.contrib.auth.password_validation.
    UserAttributeSimilarityValidator',
83      },
84      {
85          'NAME': 'django.contrib.auth.password_validation.
    MinimumLengthValidator',
86      },
87      {
88          'NAME': 'django.contrib.auth.password_validation.
    CommonPasswordValidator',
89      },
90      {
91          'NAME': 'django.contrib.auth.password_validation.
    NumericPasswordValidator',
```

```
    },
]


# Internationalization

LANGUAGE_CODE = 'en-us'

TIME_ZONE =  'Asia/Kolkata'

USE_I18N = True

USE_TZ = False


# Static files (CSS, JavaScript, Images)

STATIC_URL = 'static/'

# Default primary key field type

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'


TWILIO_ACCOUNT_SID = 'ACd74ea480d173b94b1afb7XXXXXXXXX'
TWILIO_AUTH_TOKEN = 'dc9b829e5abe94f817854XXXXXXXXX'
TWILIO_PHONE_NUMBER = 'XXXXX5XXXX'
```

Listing 6.2: *settings.py*

wsgi.py

```
"""
WSGI config for StudentEntryExit project.

It exposes the WSGI callable as a module-level variable named ``
    application``.

"""

import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'StudentEntryExit.
    settings')
```

```
13
14 application = get_wsgi_application()
```

Listing 6.3: *wsgi.py*

asgi.py

```
1 """
2 ASGI config for StudentEntryExit project.
3
4 It exposes the ASGI callable as a module-level variable named ``
    application``.
5 """
6
7 import os
8
9 from django.core.asgi import get_asgi_application
10
11 os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'StudentEntryExit.
    settings')
12
13 application = get_asgi_application()
```

Listing 6.4: *asgi.py*

urls.py

```
1 """StudentEntryExit URL Configuration"""
2 from django.contrib import admin
3 from django.urls import path
4 from pro import views
5 urlpatterns = [
6     path('admin/', admin.site.urls),
7     path('',views.getIndexPage,name="home"),
8     path('login/',views.loginView),
9     path('logout/',views.logoutView,name="logout")
10 ]
```

Listing 6.5: *urls.py*

6.2.2 App Folder

views.py

```
from django.shortcuts import render,HttpResponseRedirect
from django.utils import timezone
from django.views.decorators.csrf import csrf_exempt
from django.contrib.auth.forms import AuthenticationForm
from django.contrib.auth import authenticate,login,logout
from .models import Student
from .models import EntryExitTime

from twilio.rest import Client
from django.conf import settings
import easyocr
import base64
from PIL import Image
from io import BytesIO
import re
import datetime

@csrf_exempt
def getIndexPage(request):
    if request.user.is_authenticated:
        if request.method == 'POST':
            print(request.user.is_authenticated)
            base64_image_data = request.POST['snapshotData']
            data=base64_image_data.split(',')
            base64_image_data=data[1]
            reader = easyocr.Reader(['en'])
            image_data = base64.b64decode(base64_image_data)
            image = Image.open(BytesIO(image_data))
            result = reader.readtext(image)
            extracted_text = ' '.join([entry[1] for entry in
    result])
            print("Extracted Text:", extracted_text)
            match = re.search(r'\b[A-Z0-9]{10}\b', extracted_text)
            if match:
                extracted_number = match.group(0)
                print("Extracted Number:", extracted_number)
                try:
                    stu=Student.objects.get(id=extracted_number)
                except:
                    return render(request,"index.html",{"msg":"
    student data doesnot matched",'color':'red'})
                objs=EntryExitTime.objects.filter(stu=stu).
    order_by("-EntryTime")
                data=objs.values()
                if data:
                    if str(data[0]['EntryTime'])[:10] == str(
```

```
    datetime.datetime.now())[:10]:
44                          print(str(data[0]['ExitTime'])=="None")
45                          if str(data[0]['ExitTime'])=="None":
46                              obj=EntryExitTime.objects.filter(stu=
    Student.objects.get(id=extracted_number),EntryTime=data[0]['
    EntryTime'])
47                              obj.update(ExitTime=timezone.now())
48                              return render(request,"index.html",{"
    msg":"exited @"+str(timezone.now()),'color':"green"})
49                          else:
50                              obj=EntryExitTime(stu=Student.objects.
    get(id=extracted_number),EntryTime=timezone.now())
51                              obj.save()
52                              return render(request,"index.html",{"
    msg":"Entry @"+str(timezone.now()),'color':'green'})
53                      elif str(data[0]['EntryTime'])[:10] < str(
    datetime.datetime.now())[:10]:
54                          obj=EntryExitTime(stu=Student.objects.get(
    id=extracted_number),EntryTime=timezone.now())
55                          obj.save()
56                          return render(request,"index.html",{"msg":
    "Entry @"+str(timezone.now()),'color':'green'})
57                  else:
58                      obj=Student.objects.get(id=extracted_number)
59                      if obj:
60                          obj=EntryExitTime(stu=Student.objects.get(
    id=extracted_number),EntryTime=timezone.now())
61                          obj.save()
62                          return render(request,"index.html",{"msg":
    "Entry @"+str(timezone.now()),'color':'green'})
63                      else:
64                          return render(request,"home.html",{"msg":"
    no record found with the extracted info",'color':'red'})
65              else:
66                  return render(request,"index.html",{"msg":"no data
     is extracted",'color':'red'})
67          return render(request,"index.html")
68      else:
69          return HttpResponseRedirect('login/')
70  def loginView(request):
71      if request.method == "POST":
72          uname=request.POST['username']
73          upass=request.POST['password']
74          print(uname,upass)
75          user=authenticate(username=uname,password=upass)
76          if user is not None:
```

```
77              login(request,user)
78              return HttpResponseRedirect('/')
79          return render(request,"login.html",{'msg':'username  name
    or  password is wrong','color':'red'})
80      return render(request,"login.html")
81  def logoutView(request):
82      if request.user.is_authenticated:
83          logout(request)
84      return HttpResponseRedirect('../login/')
85
86  def send_whatsapp_message(request):
87      account_sid = settings.TWILIO_ACCOUNT_SID
88      auth_token = settings.TWILIO_AUTH_TOKEN
89      twilio_phone_number = settings.TWILIO_PHONE_NUMBER
90
91      client = Client(account_sid, auth_token)
92
93      message = client.messages.create(
94          body="Hello, this is a WhatsApp message!",
95          from_=f"whatsapp:{+1234567890}",
96          to="whatsapp:9876543210"  # Replace with the recipient's
    phone number
97      )
98
99      return render(request, 'success.html', {'message_sid': message
    .sid})
```

Listing 6.6: *views.py*

admin.py

```
1  from django.contrib import admin
2  from .models import Student,EntryExitTime
3  # Register your models here.
4
5  @admin.register(Student)
6  class AdimStudent(admin.ModelAdmin):
7      list_display=["id","name"]
8
9  @admin.register(EntryExitTime)
10 class AdminEntryExitTime(admin.ModelAdmin):
11     list_display=["stu","EntryTime","ExitTime"]
```

Listing 6.7: *admin.py*

apps.py

```
from django.apps import AppConfig


class ProConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'pro'
```

Listing 6.8: *apps.py*

6.2.3 Templates Folder

login.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="styles.css" />
    <title>Login Page</title>
    <style>
      body {
        margin: 0;
        padding: 0;
        font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
        display: flex;
        align-items: center;
        justify-content: center;
        min-height: 100vh;
        background: linear-gradient(to right, #667eea, #764ba2);
      }

      h2 {
        text-align: center;
      }

      .container {
        max-width: 400px;
        width: 100%;
      }

```

```
    .glass {
      background: rgba(255, 255, 255, 0.1);
      backdrop-filter: blur(10px);
      border-radius: 10px;
      padding: 20px;
      box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    }

    .input-group {
      margin-bottom: 20px;
    }

    label {
      display: block;
      font-size: 16px;
      margin-bottom: 5px;
      color: #fff;
    }

    input {
      width: 100%;
      padding: 8px;
      font-size: 16px;
      border: 1px solid #ccc;
      border-radius: 5px;
      background: rgba(255, 255, 255, 0.1);
      color: #fff;
      outline: none;
    }

    button {
      width: 100%;
      padding: 10px;
      font-size: 18px;
      border: none;
      border-radius: 5px;
      background: linear-gradient(to right, #667eea, #764ba2);
      color: #fff;
      cursor: pointer;
    }

    @media screen and (max-width: 600px) {
      .container {
        max-width: 100%;
      }
    }
```

```
75        p {
76          font-size: 1rem;
77          text-align: center;
78        }
79      </style>
80    </head>
81
82    <body>
83      <div class="container">
84        {%if msg%}
85        <p style="color: red">{{msg}}</p>
86        {%endif%}
87        <form class="glass" method="post">
88          {% csrf_token %}
89          <h2>Login</h2>
90          <div class="input-group">
91            <input
92              type="text"
93              id="username"
94              name="username"
95              required
96              placeholder="username"
97            />
98          </div>
99          <div class="input-group">
100           <label for="password">Password</label>
101           <input
102             type="password"
103             id="password"
104             name="password"
105             required
106             placeholder="password"
107           />
108         </div>
109         <button type="submit">Login</button>
110       </form>
111     </div>
112   </body>
113 </html>
```

Listing 6.9: *login.html*

home.html

```
1 {% include "base.html" %}
2 <html>
```

```
  <body class="pl-3">

    <h1>Student info</h1>
    <form action="http://127.0.0.1:8000/home/" method="post">
      {% csrf_token %}
      <input type="text" name="id" />
      <input type="submit" value="Fetch data" />
    </form>
    {% if msg %}
    <p style="color="green"">{{msg}}</p>
    {% endif %}
    {% comment %} {% if data%}
    <h1>Data</h1>
    <table>
      {%for dt in data%}
      <tr>
        {% for key, value in dt.items() %}
        <td>{{key}}{{value}}</td>
        {% endfor %}
      </tr>
      {% endfor %}
    </table>
    {% endif %} {% endcomment %}
  </body>
</html>
```

Listing 6.10: *home.html*

base.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Camera Snapshot with Automatic Submission</title>
    <link rel="stylesheet" href="index.css" />
    <link
      rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/
    bootstrap.min.css"
      integrity="sha384-Gn5384xqQ1aoWXA+058
    RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
      crossorigin="anonymous"
    />
  </head>
  <body class="bg-dark">
    {% block navbar %}
```

```
15      <nav
16        class="navbar navbar-expand-sm navbar-dark font-weight-bold
    bg-info"
17        id="nav"
18      >
19        <div class="container-fluid">
20          <a class="navbar-brand" href="#"> JNTUHUCES ENTRY-EXIT</a>
21          <button
22            class="navbar-toggler"
23            type="button"
24            data-bs-toggle="collapse"
25            data-bs-target="#navbarSupportedContent"
26            aria-controls="navbarSupportedContent"
27            aria-expanded="false"
28            aria-label="Toggle navigation"
29          >
30            <span class="navbar-toggler-icon"></span>
31          </button>
32          <div class="collapse navbar-collapse" id="
    navbarSupportedContent">
33            <ul class="navbar-nav ml-auto mb-2 mb-lg-0">
34              <li class="nav-item">
35                <a class="nav-link active" aria-current="page" href=
    "{% url "home" %}">Home</a>
36              </li>
37
38              {% comment %}
39              <li class="nav-item">
40                <a class="nav-link btn btn-warning text-dark" href="
    #"
41                  >Admin Login</a
42                >
43              </li>
44              {% endcomment %}
45              <li class="nav-item">
46                <a
47                  class="nav-link btn btn-warning text-dark"
48                  href="{% url 'logout' %}"
49                  >Admin Logout</a
50                >
51              </li>
52              <li class="nav-item">
53                <img src="https://pbs.twimg.com/profile_images
    /1559799667856388096/oNNEFVfk_400x400.jpg" alt="" style="
    height:50px;width:50px;border: radius 25px;position: relative;
    bottom:5px;marigin-left:5px;">
```

```
            </li>
          </ul>
        </div>
      </div>
    </nav>
    {% endblock navbar %}
  </body>
</html>
```

Listing 6.11: *base.html*

index.html

```
{% include "base.html" %}
<!DOCTYPE html>
<html>
  <head>
    <title>Camera Snapshot with Automatic Submission</title>
    <link rel="stylesheet" href="index.css" />
    <link
      rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"
      integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
      crossorigin="anonymous"
    />
    <style>
      #text-to-fade {
        font-size: 2rem;
        text-align: center;
        opacity: 1;
        transition: opacity 1s ease-in-out;
      }
    </style>
  </head>
  <body class="bg-dark">
    {% if msg %}
    <p id="text-to-fade" style="color: {{color}};">{{msg}}</p>
    {% endif %}
    <h1 class="pt-3 pl-3 text-light">
      Camera Snapshot with Automatic Submission
    </h1>
    <video id="video" class="pt-4 pl-3" autoplay></video>
    <br />
    <img id="snapshot" alt="Snapshot" style="display: none" class=
```

```
"pl-3" />
 <canvas id="canvas" style="display: none" class="pl-3"></
canvas>
 <button id="capture" class="mt-3 ml-3 btn btn-info btn-md px-5
">
   Take Snapshot and Submit
 </button>

 <!-- Form to Submit Snapshot -->
 <form id="snapshotForm" method="post" enctype="multipart/form-
data">
   {% csrf_token %}
   <input type="hidden" id="snapshotData" name="snapshotData" /
>
 </form>

 <script>
   const video = document.getElementById("video");
   const canvas = document.getElementById("canvas");
   const captureButton = document.getElementById("capture");
   const snapshot = document.getElementById("snapshot");
   const snapshotDataInput = document.getElementById("
snapshotData");
   const snapshotForm = document.getElementById("snapshotForm")
;

   let stream;

   // Get user media (camera) stream
   navigator.mediaDevices
     .getUserMedia({ video: true })
     .then((stream) => {
       video.srcObject = stream;
       stream = stream; // Store the stream for later use
     })
     .catch((error) => {
       console.error("Error accessing camera:", error);
     });

   // Capture a snapshot and submit the form
   captureButton.addEventListener("click", () => {
     const context = canvas.getContext("2d");
     canvas.width = video.videoWidth;
     canvas.height = video.videoHeight;
     context.drawImage(video, 0, 0, canvas.width, canvas.height
);
```

```
70
71          // Display the snapshot image
72          snapshot.src = canvas.toDataURL("image/jpeg");
73          snapshot.style.display = "block";
74
75          // Hide the video element
76          video.style.display = "none";
77
78          // Show the canvas element (optional)
79          // canvas.style.display = "block";
80
81          // Set the value of the hidden input to the Base64-encoded
        image data
82          snapshotDataInput.value = canvas.toDataURL("image/jpeg");
83
84          // Disable the capture button
85          captureButton.disabled = true;
86
87          // Submit the form
88          snapshotForm.submit();
89
90          // Stop the camera stream
91          if (stream) {
92            stream.getTracks().forEach((track) => track.stop());
93          }
94        });
95        //text fade away logic
96        document.addEventListener("DOMContentLoaded", function () {
97          setTimeout(function () {
98            fadeOutText();
99          }, 10000); // 30 seconds (30,000 milliseconds)
100
101         function fadeOutText() {
102           var textToFade = document.getElementById("text-to-fade")
        ;
103           textToFade.style.opacity = 0;
104         }
105       });
106     </script>
107   </body>
108 </html>
```

Listing 6.12: *index.html*

studentReg.html

```
{% include "base.html" %}
{% block body %}
<h1 class="px-3 pt-3" style="color:aqua">Register a Student</h1>
<form action="" method="post" class="px-3 pt-3">

    <table class="text-light">
        <tr class="py-2"><td><label for="sid">Regd No: </label></td>
        <td>{% csrf_token %}
            <input type="text" id="sid" name="sid"></td></tr>

        <tr class="py-2"><td><label for="sname">Name: </label></td>
        <td>{% csrf_token %}
            <input type="text" id="sname" name="sname"></td></tr>

        <tr><td><input type="submit" class="btn btn-warning"></td></tr>
    </table>
</form>
{% endblock body %}
```

Listing 6.13: *studentReg.html*

studentDataFetch.html

```
{% include "base.html" %}
{% block body %}
<h1 class="px-3 pt-3" style="color:aqua">Fetch Student Details</h1>

<form action="" method="POST" class="px-3 pt-3">
    <table class="text-light">
        <tr class="p-3"><td><label for="sid">Regd No: </label></td>
        <td>{% csrf_token %}
            <input type="text" id="sid" name="sid"></td></tr>

        <tr><td><input type="submit" class="btn btn-warning" value="Fetch Data"></td></tr>
    </table>
</form>
<table>{{stud}}</table>
{% endblock body %}
```

Listing 6.14: *studentDataFetch.html*

Chapter 7

Results

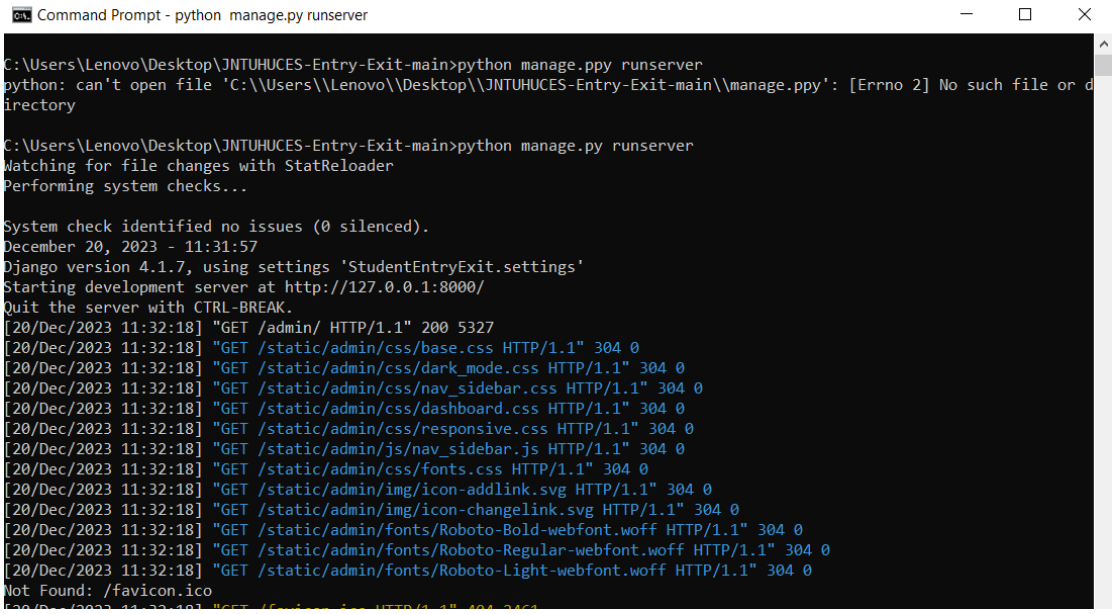

Figure 7.1: Starting Server

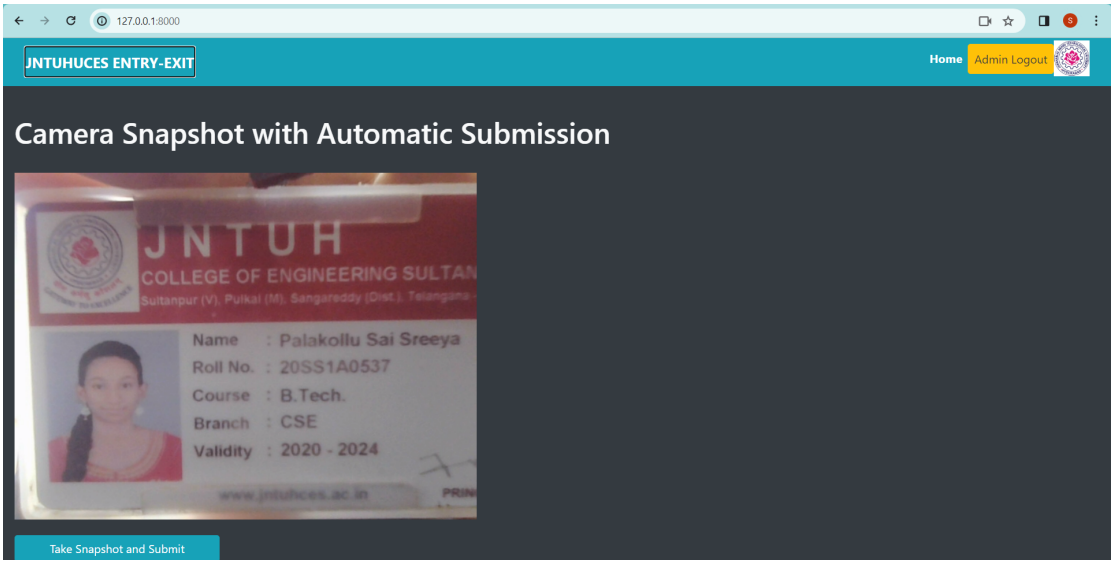

Figure 7.2: Scanning Page

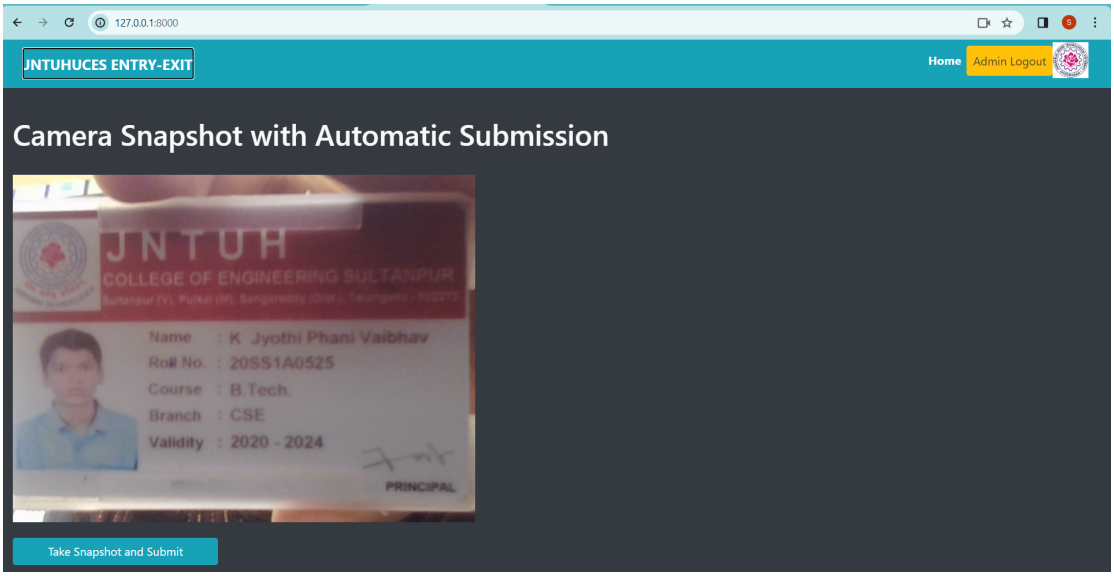

Figure 7.3: ID Card Placement

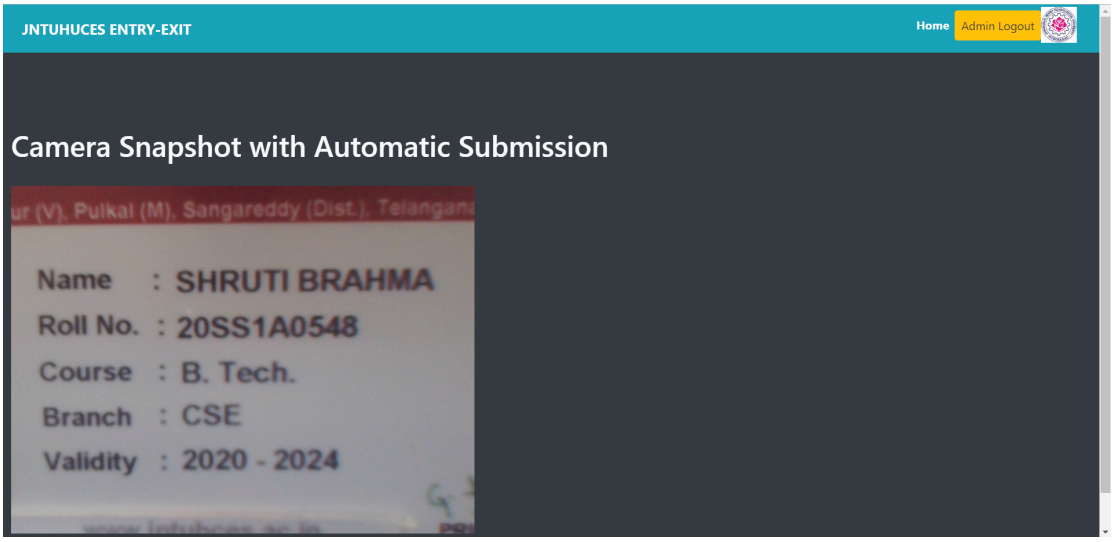

Figure 7.4: Alternative ID Card Placement

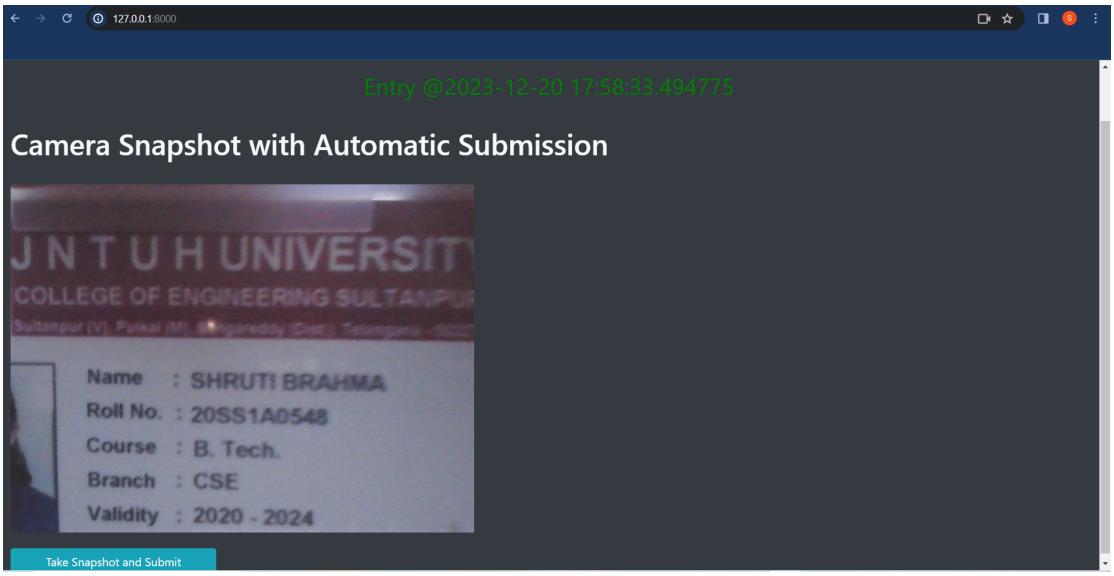

Figure 7.5: Successful Entry

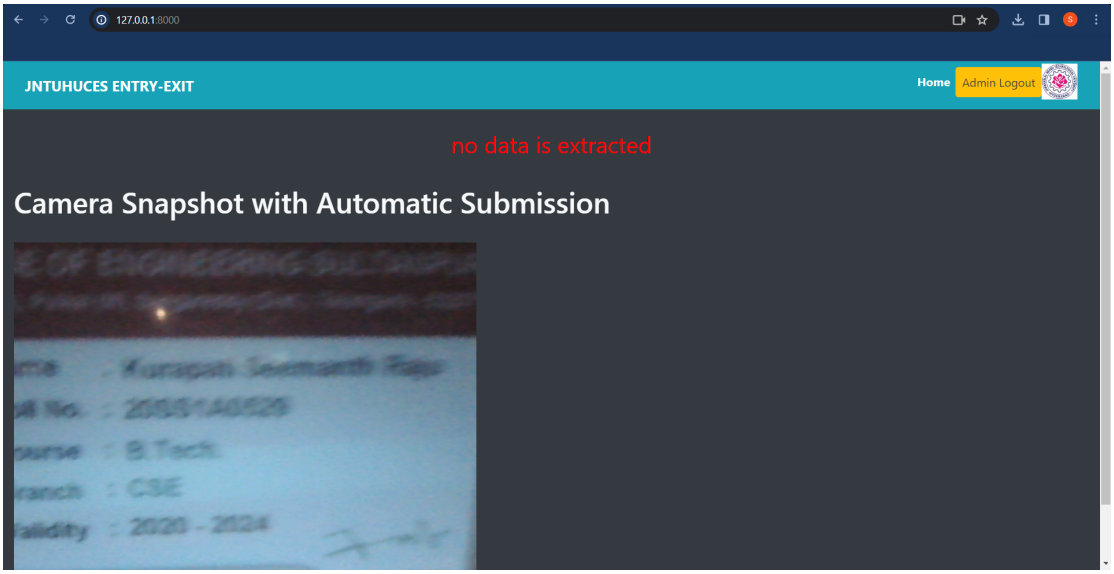

Figure 7.6: Unsucessful Scanning

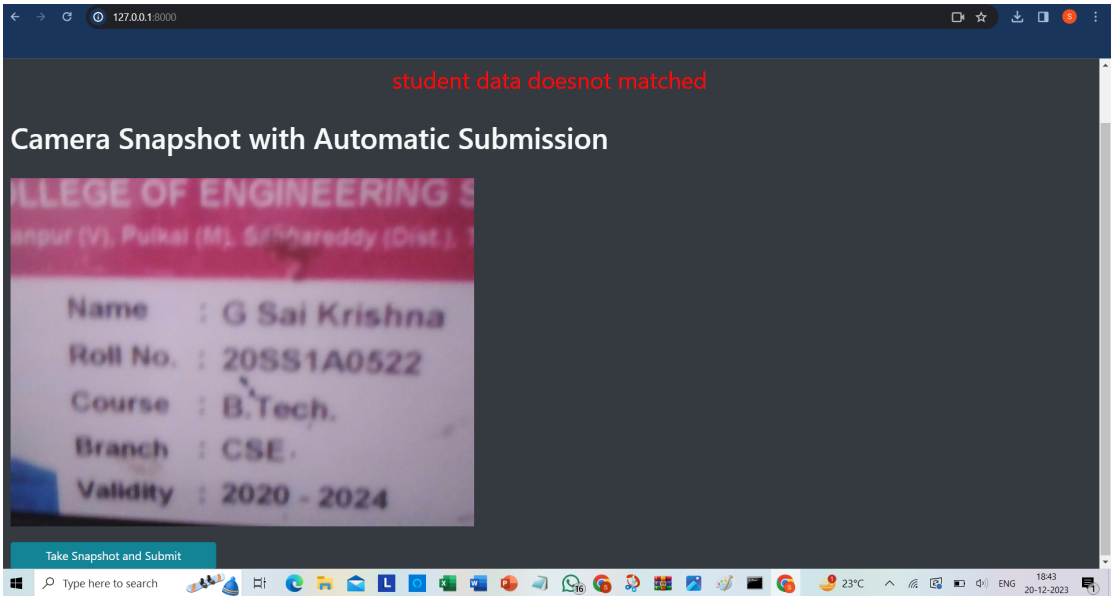

Figure 7.7: Illegal Entry/Exit

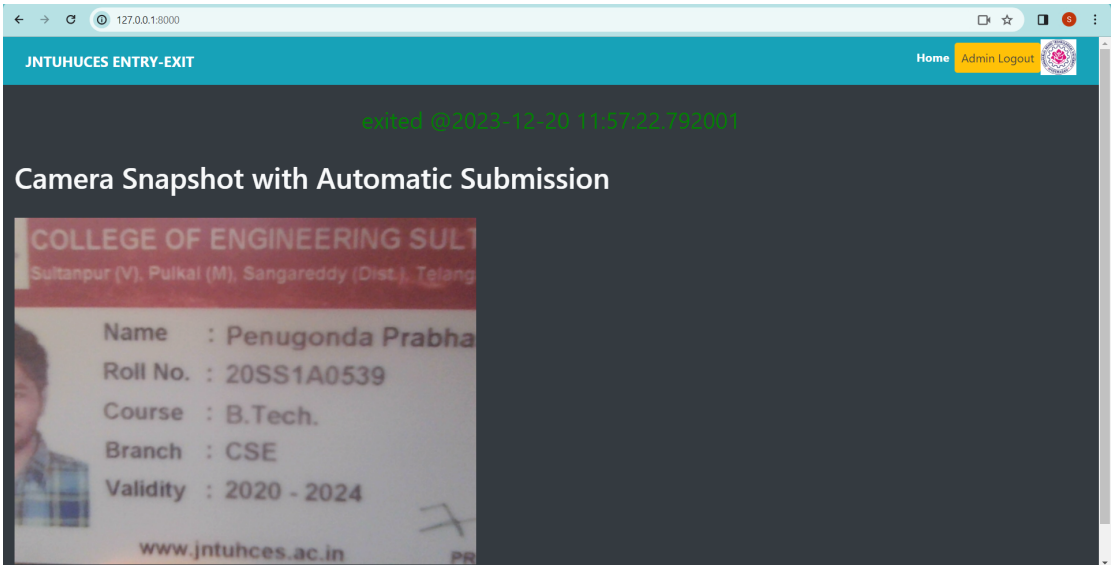

Figure 7.8: Successful Exit

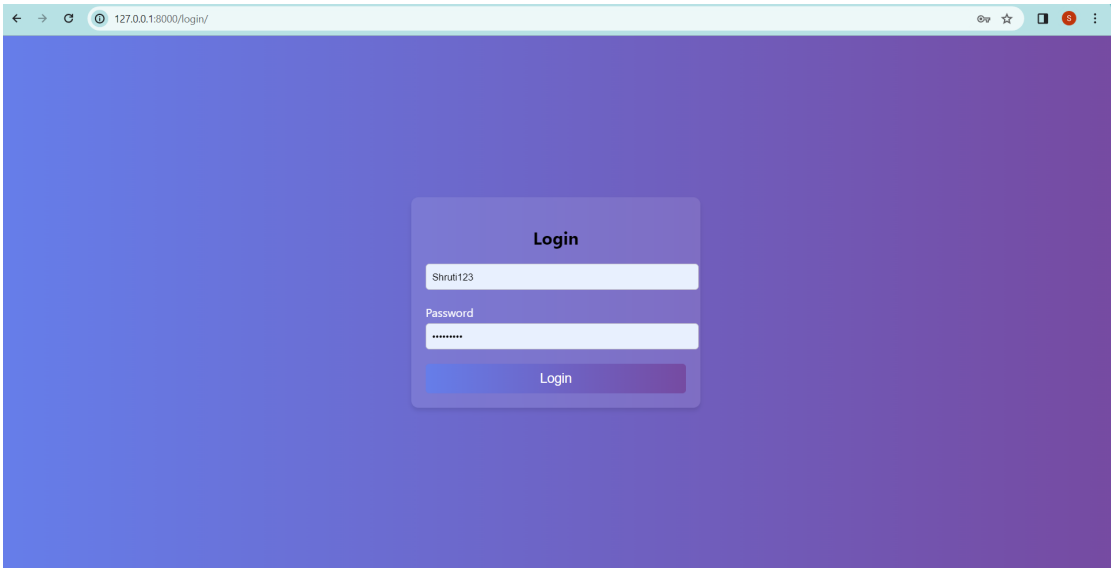

Figure 7.9: Admin Login Page

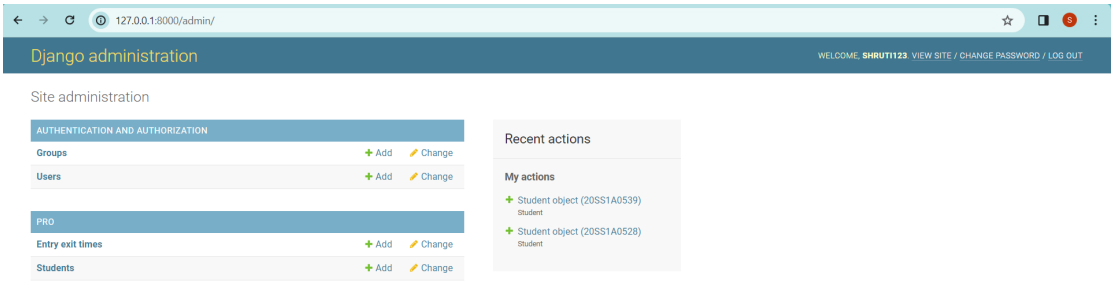

Figure 7.10: Admin Page

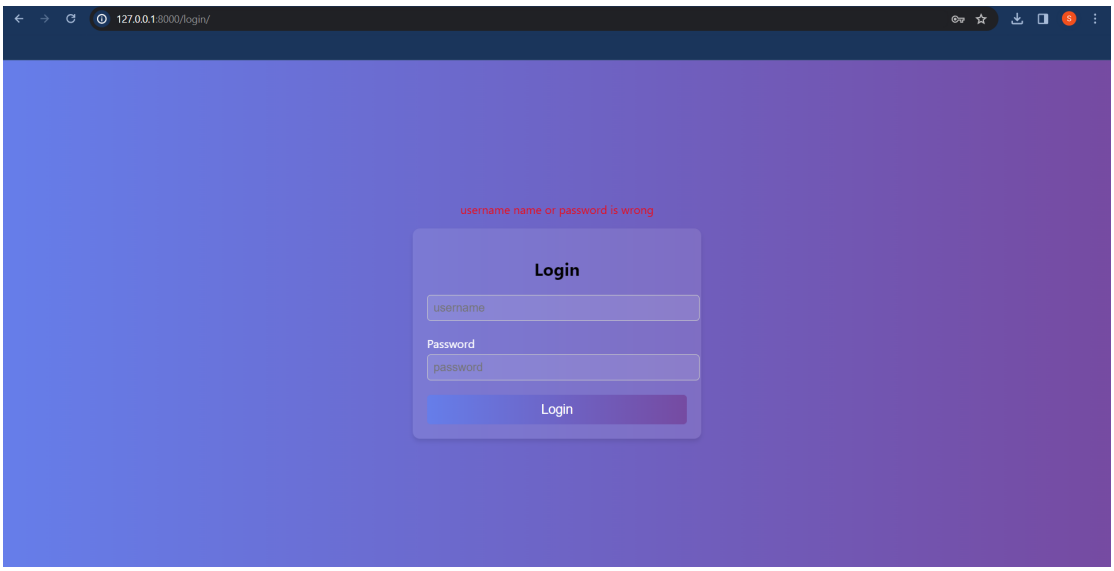

Figure 7.11: Illegal Login

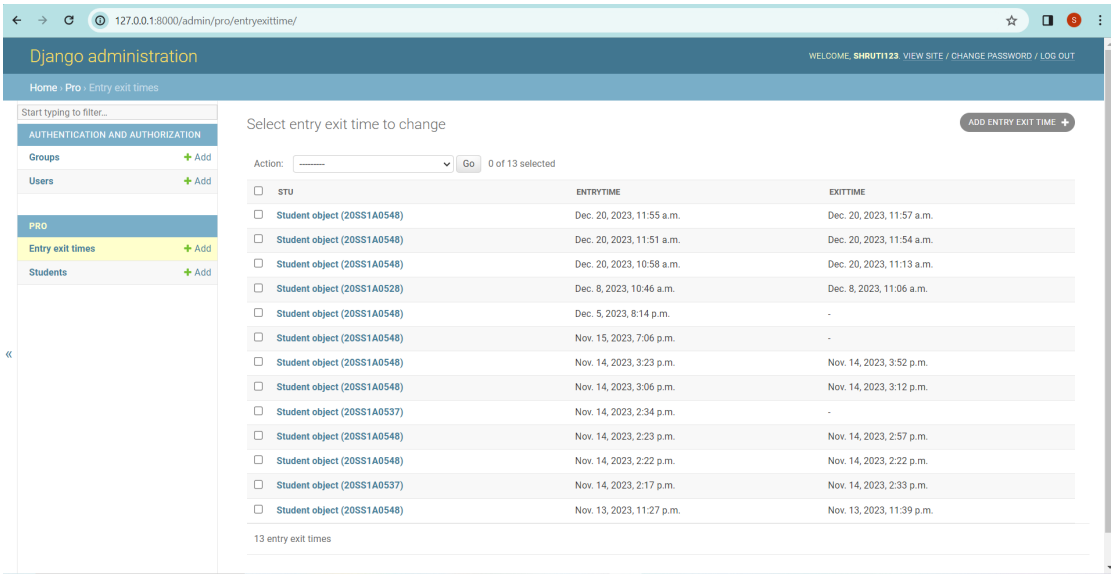

Figure 7.12: Entry-Exit Time Log

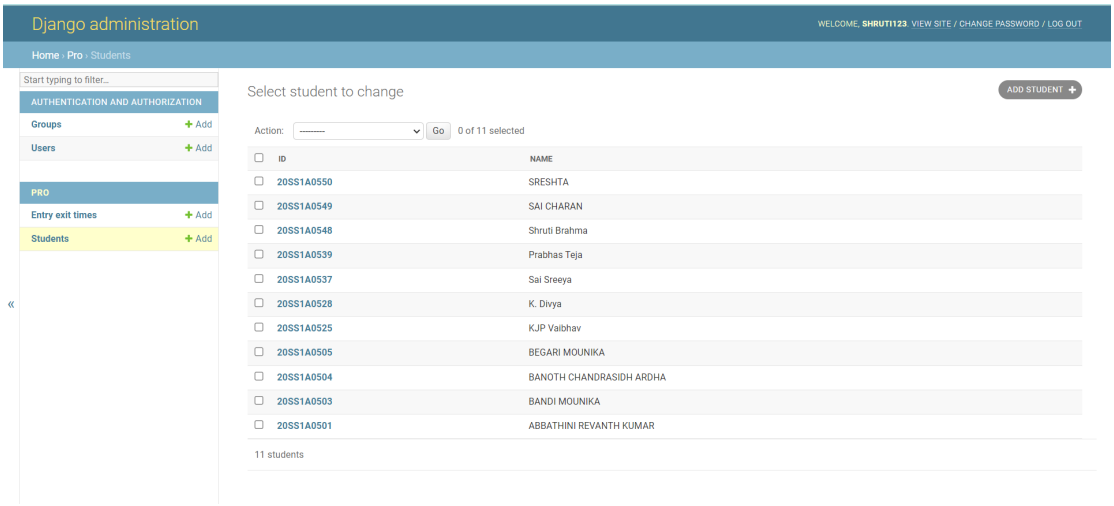

Figure 7.13: Student Database

CONCLUSION

In conclusion, the logbook system outlined in this abstract offers a sophisticated and efficient solution to the challenges associated with attendance monitoring in educational institutions. The integration of identity cards enables the seamless capture of crucial student information, such as roll number and name, coupled with precise entry and exit timestamps. This not only simplifies the attendance tracking process but also allows for the swift identification of latecomers through automated notifications.

Moreover, the system's user-friendly interface empowers both students and faculty members to easily access and review their attendance records. By fostering transparency and accountability, the logbook system contributes to a culture of punctuality among students.

The innovative approach of utilizing identity cards and implementing automated notifications demonstrates the system's commitment to providing a comprehensive solution to attendance-related challenges. As educational institutions increasingly seek ways to enhance efficiency and embrace technological advancements, this logbook system emerges as a valuable tool for streamlining attendance monitoring and promoting a more punctual and accountable learning environment.

REFERENCES

1 *Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.*

2 *Chen, Y., Wu, J., & Zhu, J. (2021). Research on Application of Location-Based Services in Campus Navigation. In 2021 10th International Conference on Software and Computing Technologies (ICSCT) (pp. 248-253). IEEE.*

3 *Da Rocha, V. M., & Rocha, H. V. (2018). Building User Interfaces with React: A Practical Guide to Components in React.js. Apress.*

4 *Farrow, R., & Bhargavan, K. (2015). Implementing TLS with Cryptographic Hardware. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15) (pp. 1403-1415). ACM.*

5 *Gupta, R., & Sharma, A. (2019). Importance of Responsive Web Design for Business. In 2019 International Conference on Communication and Electronics Systems (ICCES) (pp. 1046-1050). IEEE.*

6 *Haklay, M. (2010). How Good is Volunteered Geographical Information A Comparative Study of OpenStreetMap and Ordnance Survey Datasets. Environment and Planning B:Planning and Design, 37(4), 682–703.*

7 *Honnibal, M., & Montani, I. (2017). spaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing. To appear.*

8 *Johnson, A., Smith, M., & Anderson, P. (2019). Language Translation in Education: A Comprehensive Review. In 2019 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE) (pp. 1-6). IEEE.*

9 *Kim, H., Choi, J., Kim, J., & Kim, Y. (2018). The Effect of User Interface Design for Mobile Learning Application in Smart Learning Environment. In 2018 International Conference on Information and Communication Technology Convergence (ICTC) (pp. 1199-1201). IEEE.*

10 *Liang, D., & Chen, H. (2018). Comparative Study on Programming Languages for Software Development. In 2018 2nd International Conference on Computer Science and Application Engineering (CSAE) (pp. 355-359). IEEE.*

11 *Li, W., & Zhang, J. (2017). Design and Implementation of a Campus Intelligent Management System Based on RFID. In 2017 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2) (pp. 1-5). IEEE.*

12 *Liu, Y., & Wang, X. (2020). The Study of Language Learning Platform Based on Natural Language Processing. In 2020 IEEE 8th International Conference on Smart Energy Grid Engineering (SEGE) (pp. 269-273). IEEE.*

13 *Siekerski, K., & Manso, M. (2019). Evaluation of OpenStreetMap Data Quality at Different Stages of a Participatory Mapping Process. ISPRS International Journal of Geo-Information, 8(6), 289.*

14 *Smith, R. S., & Brown, A. (2018). CampusNav: A GPS-Based Campus Navigation App. In Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers (UbiComp '18) (pp. 862-866). ACM.*

15 *Wang, Y., & Zhang, Y. (2021). The Research of College Student Learning Behavior Analysis and Educational Model Design under the Background of Big Data. In 2021 IEEE 4th International Conference on Information and Computer Technologies (ICICT) (pp. 8-12). IEEE.*

16 *Wilson, G., Aruliah, D. A., Brown, C. T., Hong, N. P. C., Davis, M., Guy, R. T., ... & Willmore, A. (2014). Best practices for scientific computing. PLoS Biology, 12(1), e1001745*