

UNIVERSITY BUS MANAGEMENT SYSTEM

A PROJECT REPORT

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

Bachelor of Technology

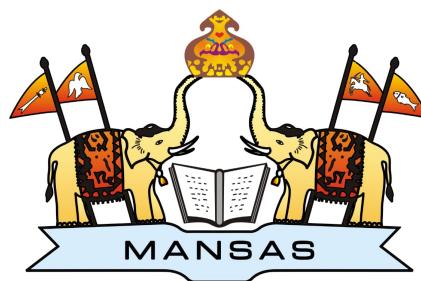
In

COMPUTER SCIENCE with IOT AND CYBERSECURITY

BY

M.KUSUMA KUMAR (22331A4735)
B. KARTHIKEYA NAIDU (22331A4705)
K. TEJA (22331A4730)
CH. SHREEKA (22331A4719)

Under the Supervision of
DR.ATCHUTA RAO SADHU
Associate Professor



**DEPARTMENT OF DATA ENGINEERING
MVGR COLLEGE OF ENGINEERING (Autonomous)**

VIZIANAGARAM- 535005, AP (INDIA)

**(Accredited by NBA, NAAC, and Permanently Affiliated to
Jawaharlal Nehru Technological University Gurajada
Vizianagaram)**

CERTIFICATE



This is to certify that the project report entitled University Bus Management System being submitted bearing M.Kusuma Kumar , B. Karthikeya Naidu, K. Teja, Ch.Shreeka roll numbers 22331A4735, 22331A4705, 22331A4730, 22331A4719 respectively, in partial fulfillment for the award of the degree of “Bachelor of Technology” in Computer Science with IOT and Cybersecurity is a record of Bonafide work done by them under my supervision during the academic year 2022-2026.

Dr. Atchuta Rao Sadhu

Associate Professor

Dept. of Data Engineering

Dr. P. Sateesh

Head of Department

Data Engineering

ACKNOWLEDGEMENTS

We place on record and warmly acknowledge the continuous encouragement, invaluable supervision, timely suggestions and inspired guidance offered by our guide **DR. Atchuta Rao Sadhu** Associate Professor, and our coordinator **MR.Vella Manikanta** Assistant Professor, Department of Computer Science with IOT and Cybersecurity MVGR College of Engineering in bringing this report to a successful completion.

We consider it our privilege to express our deepest gratitude to **Dr. P.Sateesh**, Head of the Department for his valuable suggestions and constant motivation that greatly helped the project work to get successfully completed.

DECLARATION

I hereby declare that the project entitled "**University Bus Management System**" submitted for the partial fulfilment of B. Tech Degree is my original work and the project has not formed the basis for the award of any degree or any other similar titles.

M.Kusuma Kumar
(22331A4735)

B. Karthikeya Naidu
(22331A4705)

K. Teja
(22331A4730)

Ch. Shreeka
(22331A4719)

ABSTRACT

The “University Bus Management System” project is designed to streamline and digitize the transportation services offered to students. This system provides a comprehensive platform that enables students and the transport manager to manage bus service applications, seat availability, and withdrawal processes online. Through this platform, students can view real-time information on vacant seats across different buses, check applicable fees based on their designated stops, and easily submit applications to join or withdraw from the service. Additionally, the system offers a feedback feature, allowing students to provide ratings and comments on individual buses and drivers, helping enhance the overall service quality. For the transport manager, the system simplifies the tracking of bus requests and feedback, making it easier to optimize routes and manage student inquiries. By digitizing these services, the University Bus Management System promotes a more efficient, transparent, and user-friendly transport experience for the campus community.

INDEX

Contents

| | |
|--|-----------|
| PROBLEM STATEMENT | 2 |
| INTRODUCTION | 3 |
| SYSTEM ANALYSIS | 6 |
| 1. PROBLEM DEFINITION..... | 6 |
| 2. REQUIREMENTS GATHERING | 6 |
| 3. FEASIBILITY STUDY | 6 |
| A. <i>Technical Feasibility</i> | 6 |
| B. <i>Financial Feasibility</i> | 7 |
| C. <i>Operational Feasibility</i> | 8 |
| 4. SYSTEM OBJECTIVES..... | 9 |
| 5. FUNCTIONAL SPECIFICATIONS | 9 |
| 6. PROCESS FLOW..... | 9 |
| 7. DATA ANALYSIS | 10 |
| 8. SYSTEM DESIGN CONSTRAINTS..... | 10 |
| EXISTING SYSTEM AND IT'S DISADVANTAGES..... | 11 |
| EXISTING SYSTEM | 11 |
| DISADVANTAGES OF THE EXISTING SYSTEM | 11 |
| ADVANTAGES OF PROPOSED SYSTEM..... | 12 |
| USE CASE DIAGRAM..... | 15 |
| HOW TO RUN THE PROGRAM STEP BY STEP PROCESS | 16 |
| TECHNOLOGIES USED | 18 |
| FRONT-END TECHNOLOGIES..... | 18 |
| BACK-END TECHNOLOGIES: | 18 |
| DATABASE:..... | 18 |
| DEPLOYMENT & HOSTING: | 18 |
| SOFTWARE AND HARDWARE REQUIREMENTS..... | 19 |
| CODE SNIPPETS | 20 |
| RESULTS..... | 25 |
| DATABASE TABLES..... | 33 |
| CONCLUSION..... | 34 |
| FUTURE SCOPE AND DEVELOPMENTS..... | 35 |
| REFERENCES..... | 36 |

PROBLEM STATEMENT

Managing university transportation services effectively is challenging due to the high volume of student applications, the need for real-time seat tracking, and the manual processes involved in handling requests for joining, withdrawing, and feedback submission. Traditional, paper-based methods and limited digital tracking make it difficult for the transport manager to efficiently organize routes, process applications promptly, and maintain transparency in seat allocation. Furthermore, students lack a streamlined way to access updated information on seat availability, fee structures for various stops, and the ability to submit feedback on bus services and drivers.

These inefficiencies lead to delays in processing applications, difficulty in managing high-demand bus routes, and a lack of accountability in service quality. To address these issues, a digital solution is essential to automate these processes, providing a more user-friendly experience for students and an efficient management tool for the transport administrator. The University Bus Management System seeks to resolve these problems by creating a centralized, online platform that enhances the transparency, efficiency, and quality of university transport services.

INTRODUCTION

Efficient transportation services are essential for the smooth functioning of any educational institution, especially universities with large student populations. Managing transportation services, including bus allocations, fee structures, seat availability, and handling feedback, can be complex and challenging when done manually. Traditional methods often lead to inefficiencies, delays in processing application, and lack of transparency in seat assignments and fee calculations. Additionally, students do not have easy access to current information regarding available seats, routes, or feedback channels.

The “University Bus Management System” addresses these issues by providing an integrated digital platform where students and the transport administrator can access essential transportation-related services. This system enables students to apply for bus services, check seat availability, view fee structures based on their bus stop, and submit feedback on bus services and drivers. Furthermore, it provides the transport manager with tools to process applications, manage seat assignments, and review student feedback, all in a streamlined online environment.

By introducing automation and centralization to transportation management, the University Bus Management System enhances service efficiency, improves transparency, and ultimately provides a better experience for both students and administrators. This document details the project’s objectives, features, and benefits, along with an analysis of how this system will impact the university community’s transportation experience.

Following are the modules that we have worked on:

1. Administrator Module

- Route Module
- Login
- Forms
- Updates
- Applications
- Student info

2. User Module

- Fee structure
- Vacancies
- Registration process
- Bus routes
- Feedback
- Daily updates
- Payment

1. Administrator Module

This module provides the administrator with comprehensive functionality to manage and oversee the university transportation services effectively. The administrator can add, delete, update, and view details of vehicles, routes, students, and applications. This module also facilitates the maintenance of student and faculty information for those who use the transportation service.

Route Module

The route module enables the administrator to manage details of all transportation routes available. Information such as route numbers, pickup/drop-off points, and times for each stop is maintained here. This helps ensure efficient and clear organization of all routes.

Login

The login sub-module is used to verify the authenticity of users attempting to access the system. The system includes two types of users: Admin and General Users. Each user must provide valid credentials, including username and password, to access the system.

Forms

This sub-module includes various forms for managing essential records:

- **Vehicle Information Form:** Used for entering and updating details about vehicles such as bus numbers, types, capacities, and availability.
- **Student/Faculty Registration Form:** Allows the administrator to register students and faculty for transportation services.

Updates

The updates sub-module allows the administrator to publish daily updates or announcements related to bus schedules, route changes, or service interruptions, ensuring that users are always informed.

Applications

This sub-module allows the administrator to view, approve, or reject applications submitted by students for joining or withdrawing from the bus service. The module helps track application statuses and process them efficiently.

Student Information

The administrator can view, update, and manage comprehensive information about students registered for transportation services, including their registered bus routes, payment history, and feedback records.

2. User Module

The user module is designed for students and faculty members who use the transportation services. This module provides access to essential information and features that enhance the convenience and transparency of the transportation system.

Fee Structure

This sub-module displays the transportation fee structure based on different routes and bus stops. It helps students understand the fees applicable to their specific routes and provides transparency in fee calculations.

Vacancies

This sub-module enables users to check the current seat availability on various routes. It assists students in selecting routes based on available seats and informs their decision-making for applications.

Registration Process

Users can complete their registration for bus services through this sub-module. Details such as name, email, student ID, address, and contact number are recorded in the system for the administrator's reference.

Bus Routes

Users can access detailed information about available bus routes, including route numbers, pickup/drop-off points, and timings. This helps students plan their travel effectively.

Feedback

This sub-module allows users to provide feedback on the bus service or drivers. Feedback can be reviewed by the administrator, which supports continuous service improvement.

Daily Updates

Users can view daily updates posted by the administrator, including any changes in schedules, new routes, or announcements regarding bus services. This keeps users well-informed and up-to-date.

Payment

The payment sub-module allows users to view their transportation fees and make payments online. It also provides a payment history, helping users keep track of their transactions and ensuring transparency in the fee payment process.

SYSTEM ANALYSIS

1. Problem Definition

The university currently relies on manual or outdated methods for managing its transportation services, leading to inefficiencies in handling student applications, managing bus seat allocations, and collecting feedback. This system requires a more streamlined and accessible solution that allows students and the transport manager to handle all transportation-related tasks online. Issues include:

- Delays in processing bus service applications and withdrawals.
- Lack of real-time information on seat availability and fee structure.
- No centralized way for students to provide feedback or for administrators to track feedback and application statuses.
- Inefficient communication of daily updates or schedule changes.

2. Requirements Gathering

To address the above problems, the system must meet the following requirements:

Student Requirements:

- View available seats on specific bus routes.
- Check fee structure based on chosen bus stops.
- Apply for or withdraw from the bus service.
- Provide feedback on bus services and drivers.
- View daily updates related to transportation services.
- Make payments and view payment history.

Administrator Requirements:

- Manage vehicle and route details (add, update, delete).
- Process and approve/reject student applications for joining or withdrawing from services.
- Maintain detailed student records, including transportation-related information.
- Track and review feedback to improve service quality.
- Provide daily updates or announcements for users.

3. Feasibility Study

A. Technical Feasibility

1. **Framework & Database:** Django is a powerful framework for building secure web applications, while PostgreSQL is a robust, scalable database suited for handling large datasets and complex queries.
2. **Real-Time Updates & Security:** Django supports real-time updates (e.g., seat availability) and has built-in security features. PostgreSQL adds data

encryption and access control, ensuring a secure environment for student records and payment details.

3. **Scalability:** Both Django and PostgreSQL can handle growth in user numbers and data, with cloud hosting options available for future scalability.

B. Financial Feasibility

Financial feasibility analyzes the costs involved in developing, implementing, and maintaining the University Bus Management System. Since the project will be built using Django and PostgreSQL,

both open-source tools, the initial development costs can be minimized. Below are the key financial considerations:

1. Development Costs:

- **Software and Licensing:** Django and PostgreSQL are open-source, eliminating licensing fees and reducing initial software costs.
- **Developer Salaries:** The primary development cost involves hiring skilled developers, particularly those familiar with Django and PostgreSQL, for backend and frontend work.
Additional costs may include hiring UX/UI designers to ensure user-friendly interfaces for students and administrators.
- **Integration Costs:** Integrating third-party services, such as payment gateways (e.g., Stripe, PayPal), may involve nominal costs for setup and ongoing transaction fees.

2. Infrastructure and Hosting Costs:

- **Initial Hosting:** The system can initially be hosted on a moderately priced server. Django is compatible with various hosting platforms, including cloud options such as AWS, Google Cloud, and DigitalOcean, which provide affordable, scalable hosting services.
- **Scalability:** As the number of users grows, cloud hosting allows the system to scale on demand, reducing the need for physical server upgrades. These platforms also offer managed PostgreSQL services, which simplify database maintenance and ensure high availability.
- **Ongoing Operational Costs:** Annual hosting and maintenance expenses can be estimated based on the chosen cloud provider's rates, typically ranging from a few hundred to a few thousand dollars annually, depending on user volume and data requirements.

3. Payment Processing Costs:

- **Transaction Fees:** For online payment processing (e.g., bus fees), payment gateways usually charge a small fee per transaction (typically 2-3%). This needs to be factored into the operational budget, especially as the user base grows. Using a reliable, secure payment provider ensures smooth transactions and builds user trust.

- **Financial Reporting:** Integrating a financial reporting module could streamline revenue tracking from student payments, giving the university better insights into transportation revenue and helping with budget forecasting.

4. Maintenance and Update Costs:

- **System Maintenance:** Periodic updates to Django, PostgreSQL, and the hosting environment will be necessary to maintain system security and performance. These updates can be managed by a small technical team, reducing long-term costs.
- **Support and Bug Fixes:** Allocating a budget for bug fixes and minor system enhancements ensures continuous operation and minimizes downtime. Regular backups, managed through PostgreSQL, will also incur minimal storage costs but are crucial for data integrity.
- **Future Enhancements:** As the university's transportation needs evolve, there may be costs associated with adding features, such as GPS tracking or expanded route management. Planning for incremental development ensures budget flexibility without major overhauls.

5. Training and User Support:

- **Administrator Training:** A one-time training session for the transport management team will be required. Since Django's admin panel is user-friendly, minimal training is needed, keeping this cost low.
- **Student Support:** Support resources (like a user manual or help desk) can be created to assist students, reducing the need for live technical support and keeping operational costs low.

C. Operational Feasibility

1. Ease of Use:

- **For Students:** The system is designed to be intuitive, allowing students to easily apply for services, check route availability, view updates, and make payments. The use of a web-based interface ensures accessibility from any device with an internet connection.
- **For Administrators:** The Django admin panel offers a robust backend interface, making it simple for administrators to manage vehicles, routes, applications, and user data without needing in-depth technical knowledge.

2. Efficiency Improvement: By automating the bus service application, seat allocation, feedback, and payment processes, the system eliminates paperwork and reduces the manual effort required to manage these tasks. This increases operational efficiency and reduces processing time.

3. Data Management: The system centralizes transportation-related data using PostgreSQL, providing easy access to student and vehicle information. PostgreSQL's ability to handle complex queries and large datasets simplifies record-keeping, making it easier for administrators to generate reports, track applications, and manage feedback.

4. Stakeholder Satisfaction: Students benefit from real-time updates, streamlined applications, and secure online payments, improving user satisfaction. Administrators benefit from easier data management, reducing their workload and minimizing errors associated with manual record-keeping.

5. Future Expansion: The system is designed with scalability in mind, allowing for future expansions such as adding new routes, vehicles, and advanced features (e.g., real-time GPS tracking of buses). PostgreSQL's scalability and Django's modular approach facilitate adding new functionality without overhauling the entire system.

6. Compliance: The system is built to ensure data security and privacy, in compliance with university policies and data protection regulations (e.g., GDPR). This is critical for handling sensitive information, such as student personal data and payment details.

4. System Objectives

The primary objectives of the University Bus Management System are:

- To provide a centralized, digital platform for managing university transportation.
- To enable efficient processing of applications, seat assignments, and feedback.
- To offer a transparent and accessible interface for students to manage transportation-related tasks.
- To improve communication between the transport manager and users through real-time updates and announcements.

5. Functional Specifications

The University Bus Management System will provide the following functionalities:

- **Student Functions:** Registration, view bus routes and vacancies, check fees, apply/withdraw from bus service, submit feedback, make payments, and view daily updates.
- **Administrator Functions:** Manage vehicle and route details, view and process applications, handle feedback, post daily updates, and maintain user records.

6. Process Flow

Application Process:

- Students apply for bus services through the online form, selecting their route and stop.
- Administrators receive the application, check for seat availability, and approve/reject the application.

Seat Allocation:

- Seat availability is updated in real-time as applications are approved or withdrawn. This allows students to view up-to-date information on bus capacities.

Fee Calculation:

- The system calculates the transportation fee based on the student's selected route and stop. This information is displayed during the application process and in the payment section.

Feedback Submission:

- Students can submit feedback on the bus service or drivers. The feedback is viewable by the administrator for service improvement purposes.

Daily Updates:

- Administrators can post daily updates, such as changes in bus schedules or new route information, which are visible to all registered students.

7. Data Analysis

The system will handle various data types, including:

- **User Data:** Details of students and faculty registered for bus services, including name, student ID, route details, and payment history.
- **Vehicle Data:** Information about each vehicle, such as vehicle ID, capacity, route number, and availability status.
- **Route Data:** Details about each route, including route number, stop points, and timing.
- **Application Data:** Records of each application, including application status, route choice, and seat allocation.
- **Feedback Data:** Student-submitted feedback on services and drivers, categorized by bus routes and drivers.

8. System Design Constraints

The following constraints must be considered during development:

- **Scalability:** The system should support an increasing number of users and routes as the university grows.
- **Security:** Sensitive data, such as student personal details and payment information, must be protected with appropriate authentication and encryption methods.
- **User-Friendly Interface:** Both students and administrators should find the system easy to navigate.
- **Real-Time Data:** Seat availability and updates should reflect real-time data to maintain accuracy and transparency.

EXISTING SYSTEM AND it's DISADVANTAGES

Existing System

In most universities, the transportation management system is either partially digital or entirely manual, relying on paperwork, spreadsheets, and offline communication between students and the transportation office. Students and faculty who want to register for transportation services often need to visit the transport office, fill out forms, and wait for processing. Additionally, there is limited visibility into bus schedules, seat availability, and payment processes, which leads to inconvenience and inefficiency.

Disadvantages of the Existing System

1. Inefficient Process:

- Students are often required to complete manual forms to apply for bus services, which are then manually processed by the transportation office. This results in long waiting times for approvals, updates, and other actions.
- Any changes, such as updating personal details or modifying routes, require another round of paperwork, further delaying the process.

2. Limited Access to Information:

- Students have limited access to real-time information, such as seat availability, route details, bus timings, and fees. They must visit the transport office or rely on notices, which may not always be up-to-date.
- With no centralized information system, students cannot view or track their applications or service requests online, leading to uncertainty and unnecessary follow-ups.

3. Error-Prone Data Handling:

- Since records are often maintained on paper or spreadsheets, data entry and retrieval are prone to human error. This can lead to incorrect information about bus allocations, fees, or schedules.
- Maintaining accurate, up-to-date records is challenging and requires substantial administrative work, increasing the risk of discrepancies in data.

4. Difficulty in Managing Feedback and Complaints:

- Students who wish to give feedback about the bus service or report issues (e.g., delays, driver behavior) have limited means to do so. Feedback is often not collected systematically and may be lost or ignored.
- Without a streamlined system, managing and addressing complaints is challenging for the transportation office, leading to a lack of service improvement and student dissatisfaction.

5. Lack of Real-Time Updates:

- Changes to bus routes, timings, or seat availability are not communicated in real time. This leads to confusion and inconvenience for students, especially in cases of last-minute changes or cancellations.
- The absence of real-time data means students cannot make informed decisions about bus services, affecting their overall experience and satisfaction.

ADVANTAGES OF PROPOSED SYSTEM

The proposed University Bus Management System, developed as a web-based platform using Django with a PostgreSQL database, offers numerous advantages over the existing system. By automating and digitizing transportation processes, the new system provides efficiency, convenience, and improved data management for both students and the transportation office.

1.Streamlined Application and Registration Process:

- Students can apply for bus services, update their information, or withdraw applications entirely online. This eliminates the need for in-person visits to the transport office, making the process faster and more accessible.
- Applications are processed automatically, allowing the transportation office to manage requests more efficiently and reducing wait times for students.

2.Real-Time Access to Information:

- The system provides real-time information on bus schedules, route details, seat availability, and fees, enabling students to make informed decisions.
- Updates to routes, timings, or other essential details are visible to students instantly, ensuring that they always have the latest information.

3.Enhanced Data Accuracy and Security:

- By storing all information in a centralized PostgreSQL database, data integrity is improved, reducing the chances of errors in records.
- The system includes role-based access control, which secures sensitive data. Only authorized users (e.g., administrators) can add, modify, or delete data, ensuring privacy and compliance with data protection regulations.

4.Automated Payment Processing:

- With online payment integration, students can pay transportation fees directly through the website, reducing the need for in-person transactions and manual payment tracking.
- The system automatically updates payment records, streamlining financial management for the transportation office and reducing the risk of accounting errors.

5.Improved Feedback and Complaint Management:

- Students can provide feedback on bus services or report issues (e.g., bus delays, driver behavior) directly through the platform. This feedback is logged systematically, ensuring that it reaches the relevant administrators.
- Administrators can quickly address complaints and take action based on student feedback, improving the quality of service and increasing student satisfaction.

6.Easy Access to Student and Bus Information:

- Administrators have an organized interface for managing routes, vehicles, and student registrations. They can view, add, update, or delete information effortlessly, reducing administrative workload and improving operational efficiency.

- The system's database supports the generation of reports, allowing the transportation office to make data-driven decisions (e.g., adding more routes based on demand).

7.Increased Transparency and Accountability:

- All processes, from student applications to payment status, are visible within the system. Students can track the status of their requests, ensuring transparency.
- Administrators have access to complete records of service requests, feedback, and payments, which increases accountability and provides a clear audit trail.

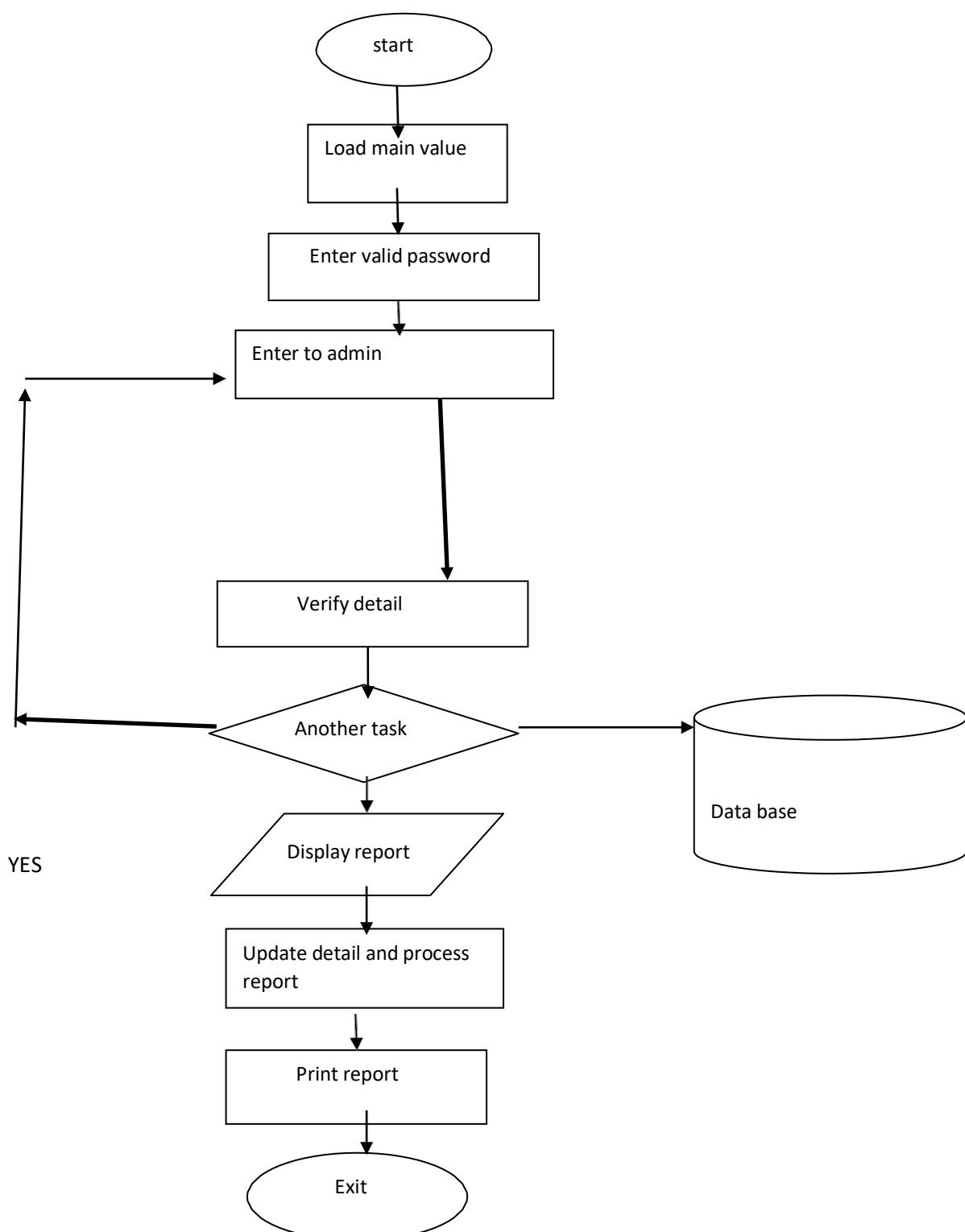
8.Scalability and Flexibility:

- The system is designed to handle an increasing number of users, routes, and vehicles, making it adaptable to future needs as the university grows.
- The modular design allows for future enhancements (e.g., GPS tracking of buses, mobile app integration) with minimal disruption to the existing system.

Summary

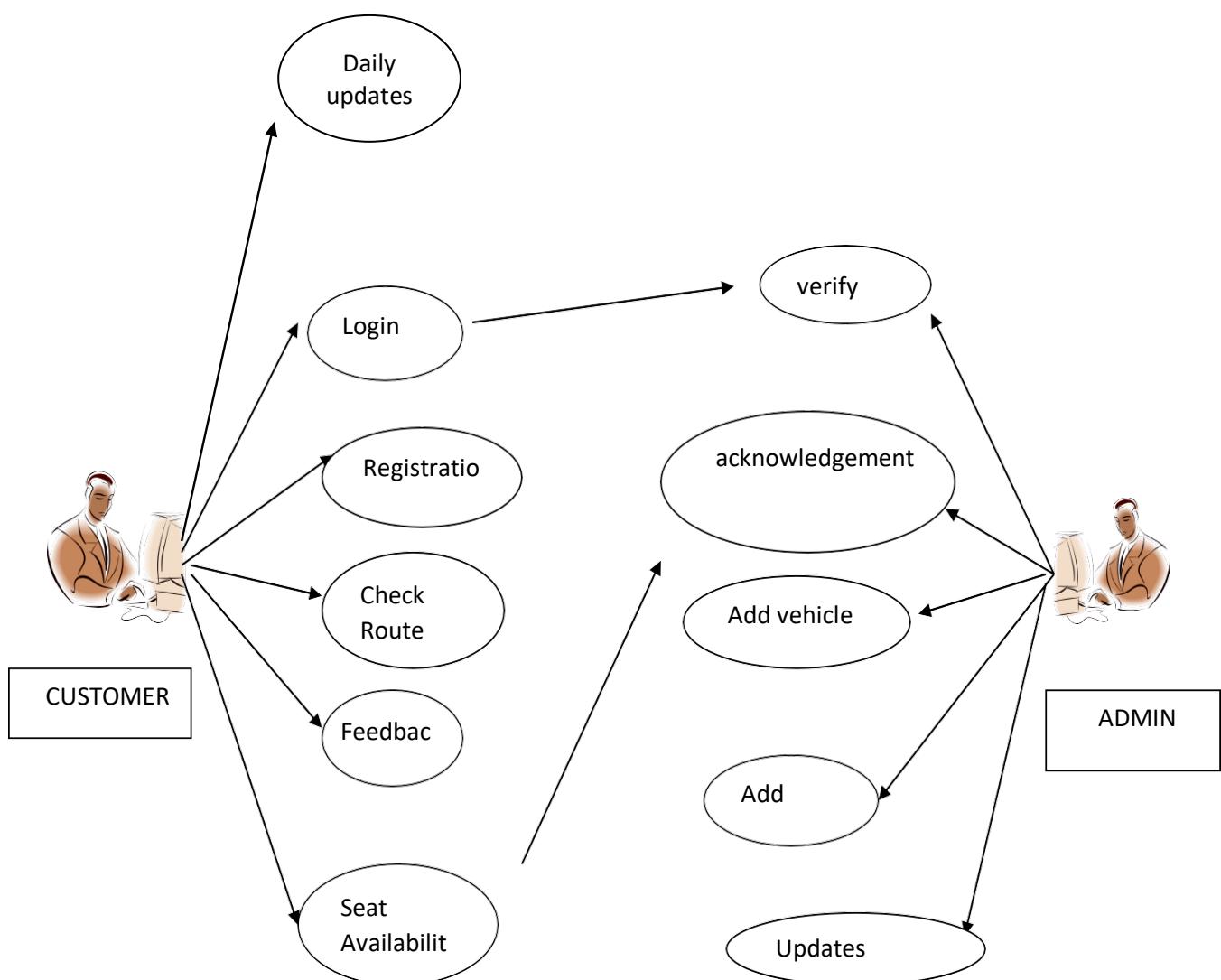
The proposed University Bus Management System delivers a modern, efficient solution that improves user experience, enhances operational efficiency, and supports data-driven decision-making. By addressing the limitations of the existing system, this platform provides a valuable tool for both students and the transportation office, fostering a more organized, accessible, and reliable transportation service.

DATA FLOW CHART



Use Case Diagram

The image shown below is a **Use Case Diagram** for the University Bus Management System. It visually represents the interactions between the **Customer (Student)** and **Admin** roles within the system. Each oval represents a distinct action or functionality that the user can perform, which helps illustrate how different users interact with the system's features.



How to run the program step by step process

Django Setup

Make sure Python is installed in our system and path setup is completed

To check python installed in our system type python --version

1.Installation of virtual environment:

Windows/macOs:

Command- python -m pip install virtualenv

Linux:

Command- sudo apt-get update

Command- sudo apt-get install python-virtualenv

2.Creation of virtual environment:

Windows:

Note

1.Check virtualenv -version if return version follow first command else second command.

2.replace projectenv with any name.Command 1- virtualenv projectenv

Command 2 if command 1 not work

Command 2- python -m venv projectenv

Macos:

Command- virtualenv projectenv

Linux:

Command- python -m venv projectenv

3.activation of virtual environment:

Windows:

Command- cd projectenv\bin

Command- activate

Macos / Linux:

Command- source projectenv/bin/activate

4.installation of Django:

Note: make user environment is active check whether looks like this or not on left side
(projectenv)

All operating systems:

Command- pip install Django

5. Creation of project:

Windows:

note

1. Make sure you're not in the environment directory if you're in that environment directory come back to original path.

2. Come back by using cd ..

3. Replace project with any name

Command-Django-admin startproject project

Macos/Linux:

Command- Django-admin startproject project

6. Creation of app:

All operating systems:

Command- cd project

Command- Django-admin startapp project app

7. Deployment:

All operating systems:

Command- python manage.py runserver

Technologies Used

Front-End Technologies:

- HTML, CSS: Used to structure and style the content of the web application, providing a user-friendly interface for students and administrators.
- JavaScript: Adds interactivity to the website, enabling dynamic content updates without reloading the page.

Back-End Technologies:

- Django: A powerful Python-based web framework used to build the application's server-side logic. Django manages data flow, handles requests, and connects the front-end to the database.

Database:

- PostgreSQL: An open-source relational database used to store and manage all application data, including user details, bus routes, schedules, and feedback. PostgreSQL supports complex queries, ensuring efficient data handling for the application.

Deployment & Hosting:

- Heroku / AWS / DigitalOcean: Possible hosting solutions for deploying the Django application. These platforms support scalability and allow continuous deployment, ensuring the system is accessible to users anytime.
- Git/GitHub: Version control is managed through Git, with GitHub used for source code management, collaboration, and deployment automation.

This technology stack ensures the University Bus Management System is efficient, scalable, and user-friendly, meeting the needs of both students and administrators.

Software and Hardware Requirements

Software Requirements:

Operating System: Linux, Windows, or macOS (for development and deployment).

Web Browser: Chrome, Firefox, Safari (for the web-based application).

Programming Languages: JavaScript (React/Node.js), Python, or other back-end languages.

Database Software: PostgreSQL

Development Frameworks: Django.

Hardware Requirements:

Client Devices: Any computer, tablet, or smartphone with internet connectivity to access the web or mobile application.

Server Hardware: Virtual servers in cloud platforms or physical servers with sufficient memory (8 GB RAM or higher) and storage (100 GB or higher) to store visitor data and run the system smoothly.

Network Infrastructure: High-speed internet connection for real-time access and data synchronization

Code Snippets

Index.html

```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>University Bus Management System</title>
    <link rel="stylesheet" href="{% static 'styles.css' %}">
    <script src="{% static 'script.js' %}" defer></script> <!-- Include script.js -->

</head>
<body>
    <div>
        <marquee behavior="scroll-animation" direction="left" class="marquee">MVGR
College of Engineering</marquee>

        <header>
            <div class="logo">
                
            </div>

            <div class = "header-content">
                <h1>College Transportation </h1>
                <nav>
                    <ul>
                        <li><a href="{% url 'index' %}">Home</a></li>
                        <li><a href="{% url 'routes' %}">Bus Routes</a></li>
                        <li><a href="{% url 'login' %}">Student Login</a></li>
                        <li><a href="{% url 'login' %}">Staff Login</a></li>
                        <li><a href="{% url 'apply' %}">Application Form</a></li>
                        <li><a href="{% url 'contact' %}">Contact</a></li>
                    </ul>
                </nav>
            </div>
        </header>
    </div>

    <!-- Container for Main Content -->
    <div class="container">
        <section id="home">
            <h1>Welcome to the University Bus Management System</h1>
            <p>Manage your bus schedules and applications with ease.</p>

            <!-- Carousel Section -->
            <div class="carousel-container">
```

```

<div class="carousel-background"></div>
<div class="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <button class="carousel-control prev" onclick="prevSlide()">&#10094;</button>
  <button class="carousel-control next" onclick="nextSlide()">&#10095;</button>
</div>
</div>
</section>

<!-- Scroll Information Section -->
<section id="info" class="scroll-animation">
  <h2>For Communication</h2>
  <div class="info-content">
    
    <div class="info-text">
      <h3>Dr. ATCHUTA RAO SADHU</h3>
      <p>Associate Professor, DATA Engineering.<br>
        CONVENOR - COLLEGE TRANSPORT <br>
        Mobile: 9441159714<br>
        Email : transportation.mvgr@gmail.com.
      </p>
    </div>
  </div>

  <h2>Total Buses Information</h2>
  <p>We currently operate <strong>25 buses</strong> covering various routes
throughout the city to ensure safe and reliable transportation for our students and
staff.</p>
</section>

<!-- Additional Transportation Details Section -->
<section class="transportation-details">
  <h2>Transportation Details</h2>
  <p>Our transportation system includes a variety of services to ensure comfort and
efficiency for all users. We provide:</p>
  <ul>
    <li>Regularly scheduled bus services throughout the week.</li>
    <li>On-demand transport options for special requirements.</li>
  </ul>
</section>

```

```

<li>Real-time tracking for all buses via our mobile app.</li>
<li>Special arrangements for events and emergencies.</li>
</ul>
</section>
</div>

<footer>
<p>&copy; 2024 University Bus Management System</p>
<p>Developed by: M.KusumaKumar B.karthikeya Teja Shreekha</p>
</footer>
</body>
</html>

```

models.py

```

from django.db import models
from django.utils import timezone
from django.contrib.auth.models import User

class Bus(models.Model):
    name = models.CharField(max_length=100) # Bus name
    routes = models.TextField() # List of stops
    morning_start_time = models.TimeField() # Start time in the morning
    college_start_time = models.TimeField() # Start time from the college
    description = models.TextField(blank=True, null=True) # Optional description field
for notes
    image = models.ImageField(upload_to='media', blank = True, null=True) # Upload
field for images

    def __str__(self):
        return self.name

class BusImage(models.Model):
    image = models.ImageField(upload_to='media', blank = True, null=True) # Image field
    note = models.TextField(blank=True, null=True) # Additional description or note

    def __str__(self):
        return f'Image {self.id}'

class BusApplication(models.Model):
    full_name = models.CharField(max_length=100)
    student_id = models.CharField(max_length=20)
    contact_number = models.CharField(max_length=15)
    email = models.EmailField()
    route = models.CharField(max_length=50)
    bus_stop = models.CharField(max_length=100)
    Section = models.CharField(max_length=100)
    created_at = models.DateTimeField(default=timezone.now)

```

```

def __str__(self):
    return self.student_id

# models.py

class Student(models.Model):
    student_id = models.CharField(max_length=20, unique=True)
    full_name = models.CharField(max_length=100)
    course = models.CharField(max_length=100)
    year = models.IntegerField()
    image = models.ImageField(upload_to='media', blank=True, null=True)
    fee_paid = models.BooleanField(default=False)
    balance_amount = models.DecimalField(max_digits=10, decimal_places=2,
default=0.00)
    semester_fee = models.DecimalField(max_digits=10, decimal_places=2,
default=500.00)
    transport_fee = models.DecimalField(max_digits=10, decimal_places=2,
default=200.00)
    paid_amount = models.DecimalField(max_digits=10, decimal_places=2, default=0.00)

    def __str__(self):
        return f'{self.full_name} ({self.student_id})'

class StudentLogin(models.Model):
    student = models.OneToOneField(Student, on_delete=models.CASCADE)
    password = models.CharField(max_length=50) # This should be hashed for security

    def __str__(self):

```

views.py

```

from django.shortcuts import render, redirect
from django.contrib.auth import authenticate, login as auth_login
from django.contrib.auth.decorators import login_required
from django.http import HttpResponseRedirect
from .models import Bus, BusImage, BusApplication
from .forms import BusApplicationForm
from django import forms
from .models import Student, StudentLogin
from .forms import StudentLoginForm

# Create your views here.

def index(request):
    return render(request, 'index.html')
def routes(request):

```

```

buses = Bus.objects.all() # Fetch all buses from the database
image = BusImage.objects.all()
return render(request, 'routes.html', {'buses': buses , 'images' : image })
def login(request):
    if request.method == 'POST':
        form = StudentLoginForm(request.POST)
        if form.is_valid():
            student_id = form.cleaned_data['student_id']
            password = form.cleaned_data['password']

            try:
                # Fetch the student login record
                student_login = StudentLogin.objects.get(student__student_id=student_id,
                password=password)
                # Redirect to the student's page
                return redirect('studentPage', student_id=student_login.student.student_id)
            except StudentLogin.DoesNotExist:
                # If login details are invalid, return an error message
                form.add_error(None, 'Invalid Student ID or Password.')
    else:
        form = StudentLoginForm()
    return render(request, 'login.html', {'form': form})

def apply(request):
    submitted = False
    if request.method == 'POST':
        form = BusApplicationForm(request.POST)
        if form.is_valid():
            form.save() # Save the form data to the database
            print("Form saved successfully.") # Debugging line
            return HttpResponseRedirect('/apply?submitted=True') # Redirect to indicate
            submission success
        else:
            print(form.errors) # Print form errors for debugging
    else:
        form = BusApplicationForm()
        if 'submitted' in request.GET:
            submitted = True

    return render(request, 'apply.html', {'form': form, 'submitted': submitted})
def contact(request):
    return render(request, 'contact.html')

def studentPage(request,student_id):
    # Fetch the student's details using the student_id
    try:
        student = Student.objects.get(student_id=student_id)
    except Student.DoesNotExist:
        return HttpResponseRedirect("Student not found", status=404)
    return render(request, 'studentPage.html', {'student': student})

```

Results

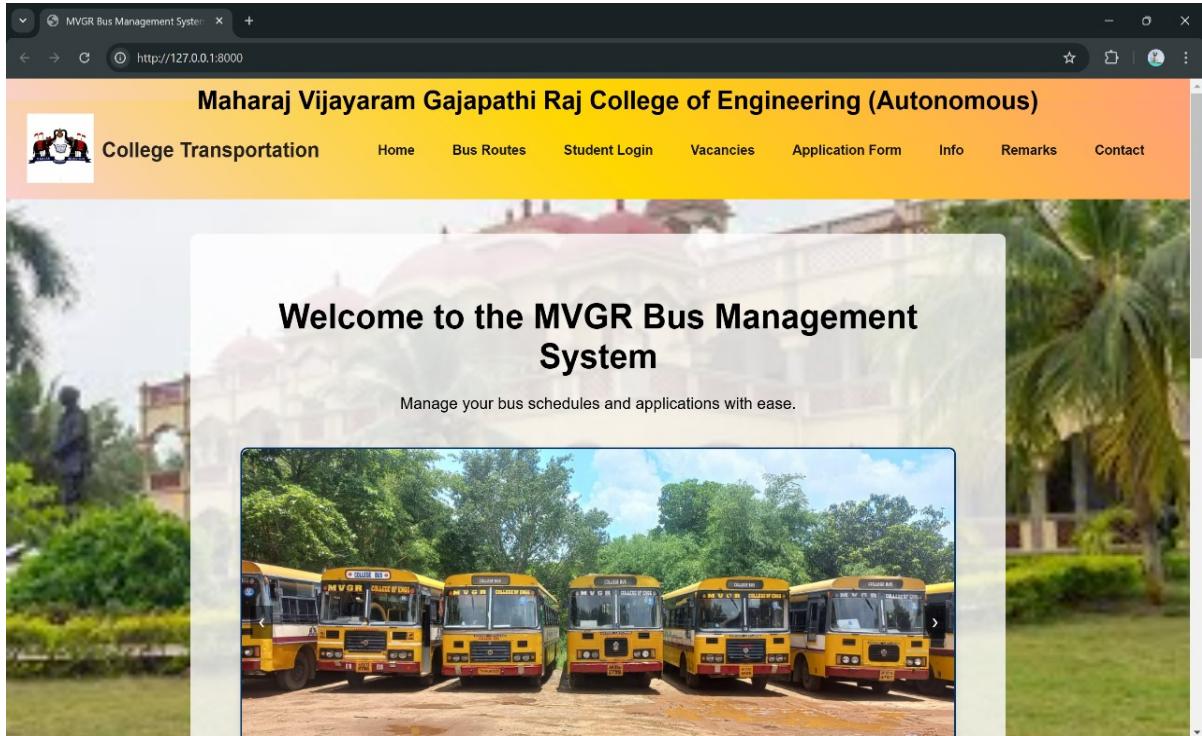


Fig.1 Home page

A screenshot of a web browser showing the "Bus Vacancies - University Bus" page. The title bar says "Bus Vacancies - University Bus" and the URL is "http://127.0.0.1:8000/vacancy/". The header features the college logo and the text "Vacancy". Below the header, there are navigation links: Home, Bus Routes, Student Login, Vacancies, Application Form, Info, Remarks, and Contact. The main content area is titled "Bus Vacancy Information" and contains a table showing bus names and their seat status. The table has columns: Bus Name, Total Seats, Occupied Seats, and Remaining Seats. The data is as follows:

Fig.2 Vacancy page

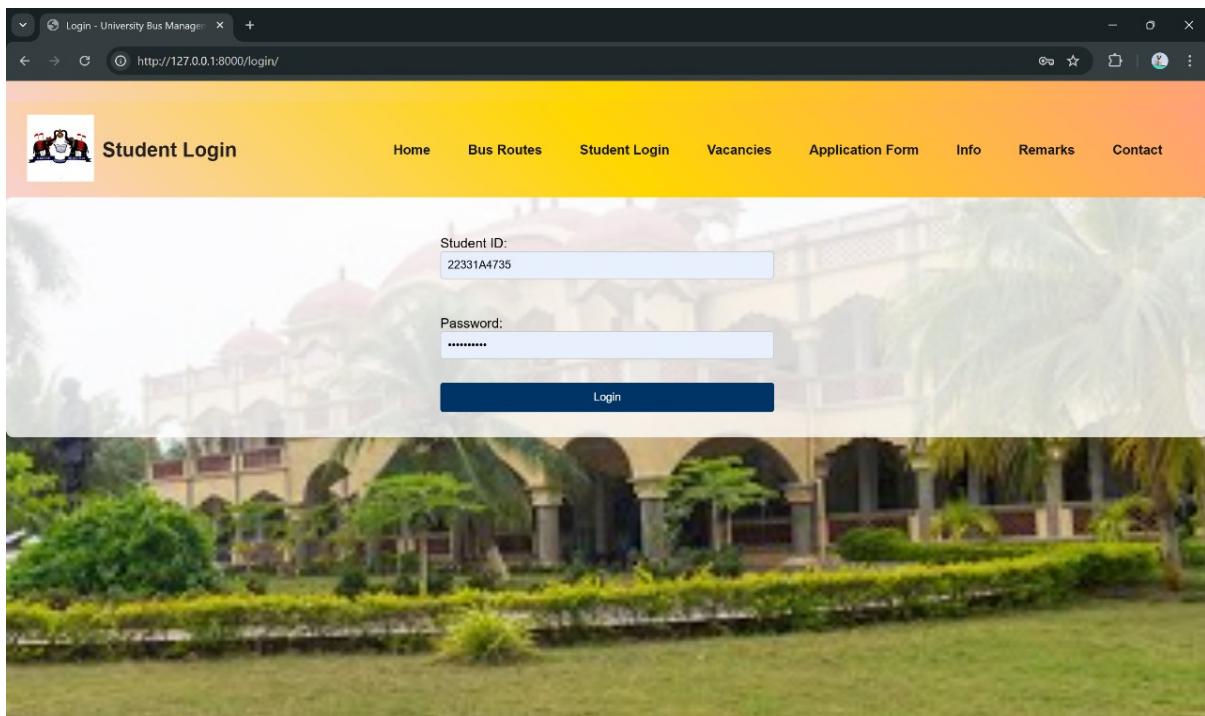


Fig.3 Student Login page

A screenshot of a web browser showing the 'Student Dashboard' page. The header is orange with the title 'Student Dashboard'. Below the header is a large background image of a university campus. The dashboard displays student details: Name - M.Kusumakumar, ID - 22331A4735, Course - CIC, Year - 3. It also shows a section titled 'Transport Fee Status' with a balance amount of ₹25000.00 and a table of fee details for each year. The table has four columns: Year, Transport Fee, Paid Amount, and Balance Amount. The data is as follows:

Transport Fee Status

Balance Amount: ₹25000.00

Fee Details for Each Year

| Year | Transport Fee | Paid Amount | Balance Amount |
|------|---------------|-------------|----------------|
| 1 | ₹35000.00 | ₹35000.00 | ₹0.00 |
| 2 | ₹35000.00 | ₹35000.00 | ₹0.00 |
| 3 | ₹35000.00 | ₹10000.00 | ₹25000.00 |

[Pay Balance](#)

Fig.4 Student Interface page

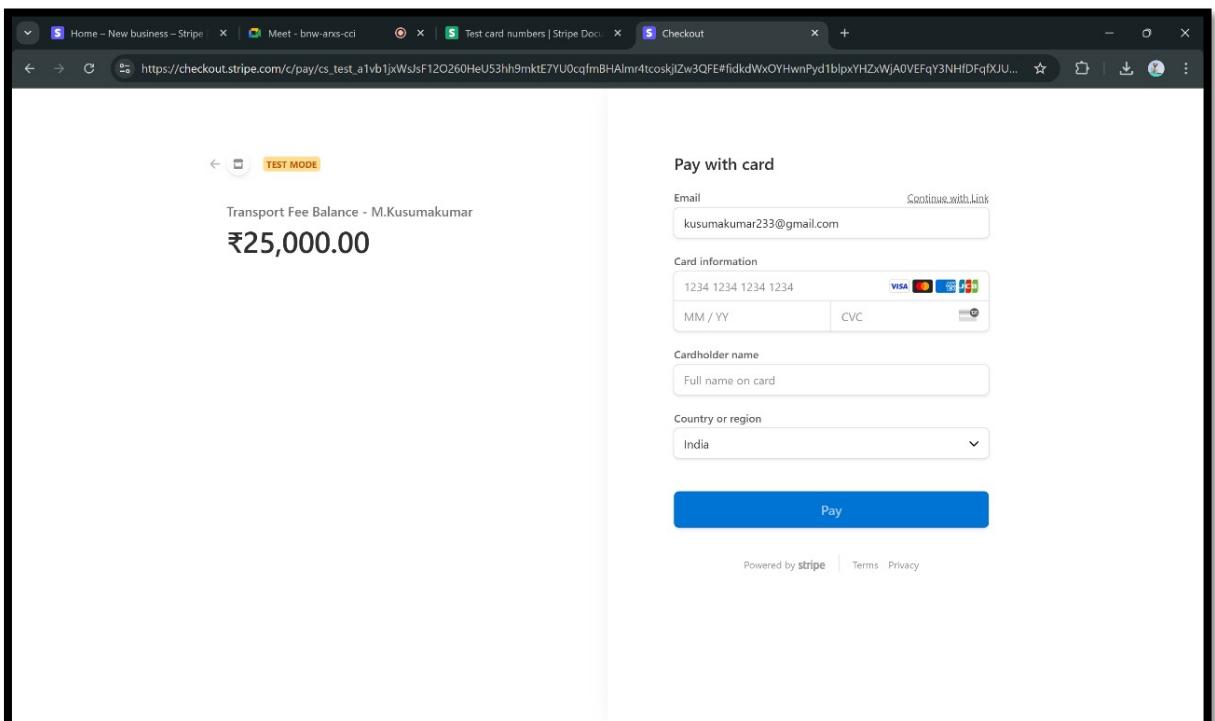
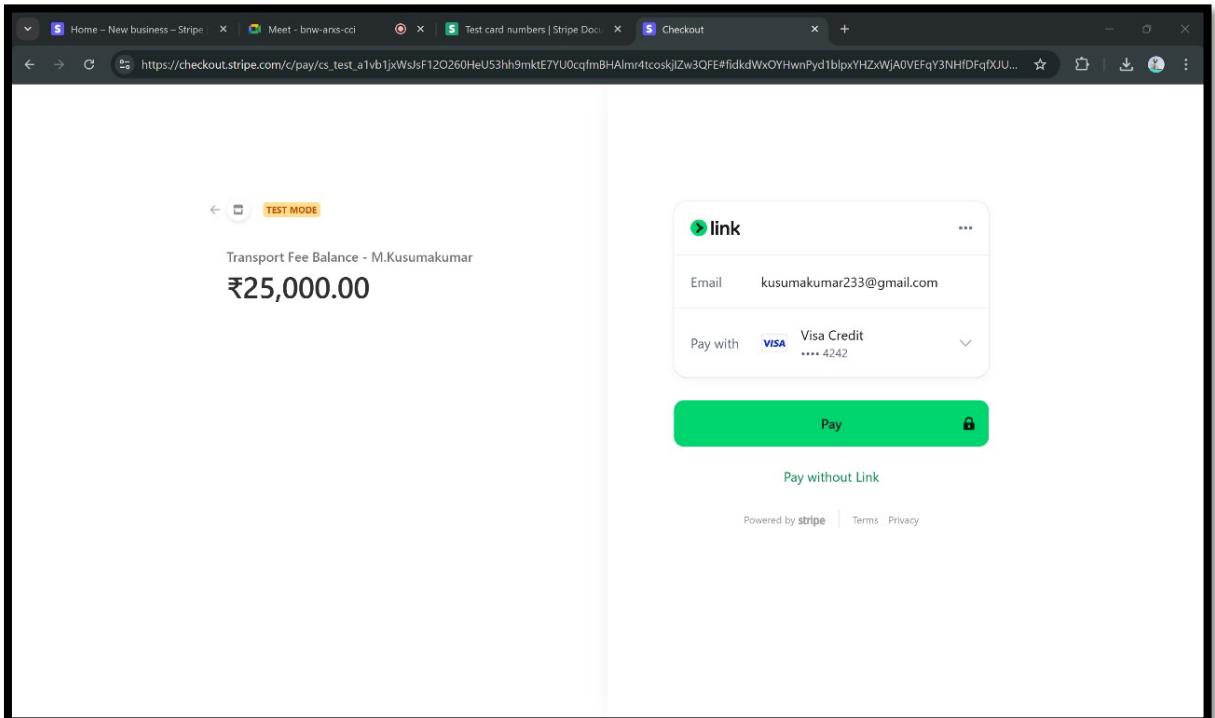


Fig.5&6 Payment pages

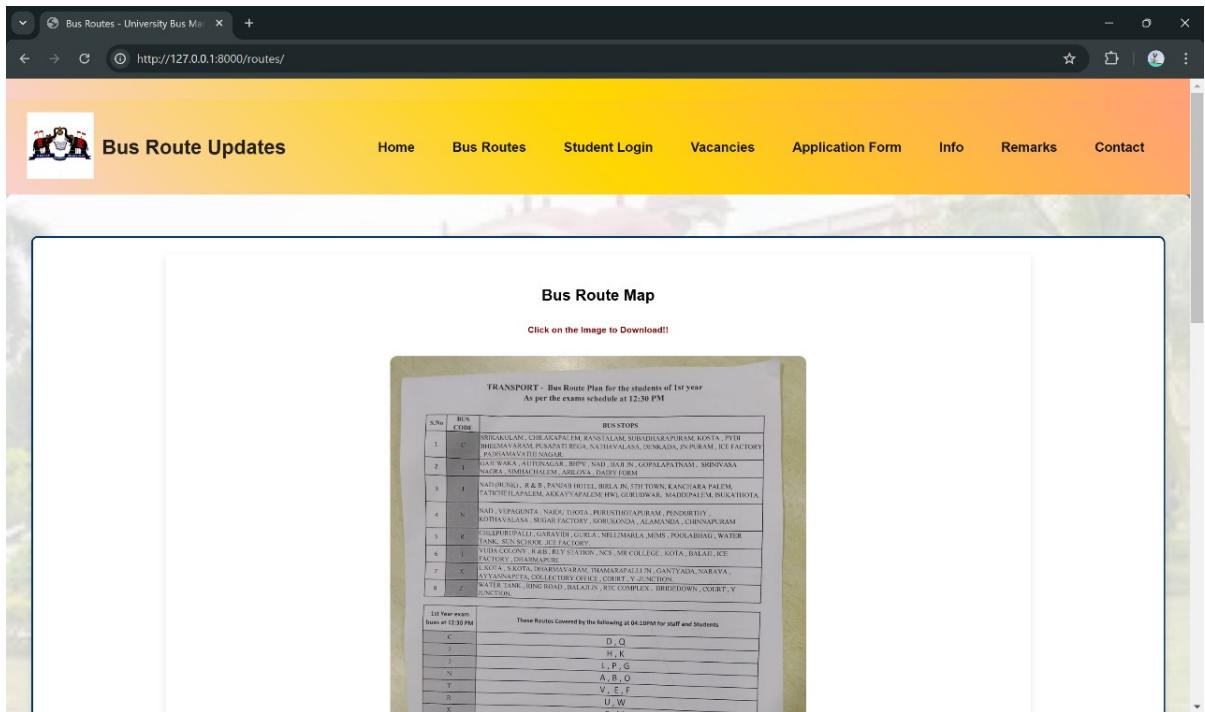


Fig.7 Update and Routes page

| BUS ROUTE MAP (Stop wise fee) 2024-25 :: VSP , SKLM , BOBBILI , S.KOTA , CHEEPURUPALLI & VZM | | | | | | | | | | | | | | | | |
|--|----------------------------------|--|-------------------------------------|----------------------------|-----------------------------|--------------------------------|---|--------------------------------|---|-------------------------------|--------------------------------------|-------------------------------|-----------------------------------|------------------------------------|---|-----------------------------|
| Bus Code | Stops | | | | | | | | | | | | | | | |
| | stop - 1 | stop - 2 | stop - 3 | stop - 4 | stop - 5 | stop - 6 | stop - 7 | stop - 8 | stop - 9 | stop - 10 | stop - 11 | stop - 12 | stop - 13 | stop - 14 | stop - 15 | stop - 16 |
| A (SENIORS) KURMANAPALEM Starts at 6:45am | Kurmannapalem (Rs.3750/-) | Vedapudi (Rs.3750/-) | Srinagar (Rs.3750/-) | Gejuwaka P 5 (Rs.3750/-) | R.K.Hospital (Rs.3750/-) | Panthulu Garl Meda (Rs.3750/-) | BHPV (Rs.3750/-) | Sheela nagar (Rs.3750/-) | Airport (Rs.3750/-) | Kakaninagar (Rs.3750/-) | LG Polymeries (Rs.3450/-) | Punshotampuram (Rs.3450/-) | Naidu Thota (Rs.3450/-) | Akkireddyapalem (Rs.3450/-) | Anandspuram (Rs.2500/-) | Rajapulova (Rs.2500/-) |
| B (JUNIORS) (Schidia) Starts at 6:45am | Scindia (Rs.3750/-) | Sriharipuram (Rs.3750/-) | ■ Gejuwaka Depo (Rs.3750/-) | BC Road (Rs.3750/-) | New Gejuwaka In (Rs.3750/-) | Old Gejuwaka In (Rs.3750/-) | ■ Sheela nagar (Rs.3750/-) | Airport (Rs.3750/-) | Gopalapatnam P 5 (Rs.3450/-) | LG Polymeries (Rs.3450/-) | Naidu Thota (Rs.3450/-) | Vepaguda (Rs.3450/-) | Sujathanagar (Rs.3450/-) | Akkireddyapalem (Rs.3450/-) | Anandapuram (Rs.2500/-) | ■ |
| C (SENIORS) (SRIKAKULAM) Starts at 6:45am | Arasavilli (Rs.3700/-) | Mill Junction (Rs.3700/-) | OBS (Rs.3700/-) | ■ Day & Night (Rs.3700/-) | PN Colony (Rs.3700/-) | ■ Kintara Mill (Rs.3700/-) | Chikalapalem (Rs.3700/-) | ■ Subhadrapuram (Rs.3400/-) | Ranastalam (Rs.3100/-) | Pathivedapalem (Rs.3100/-) | Pydhimavaram (Rs.3100/-) | Kandivalasa Gedda (Rs.3100/-) | Mylen Company (Rs.3100/-) | Poosapati Rega (Rs.2700/-) | Natalawala (Rs.2700/-) | Denkada (Rs.2700/-) |
| D (SENIORS) (SRIKAKULAM) Starts at 6:45am | Arasavilli (Rs.3700/-) | Mill Junction (Rs.3700/-) | Suryamahal (Rs.3700/-) | ■ Day & Night (Rs.3700/-) | ■ Kintara Mill (Rs.3700/-) | Chikalapalem (Rs.3700/-) | ■ Subhadrapuram (Rs.3400/-) | Ranastalam (Rs.3100/-) | Pydhimavaram (Rs.3100/-) | Kandivalasa Gedda (Rs.3100/-) | Mylen Company (Rs.3100/-) | Poosapati Rega (Rs.2700/-) | Natalawala (Rs.2700/-) | Denkada (Rs.2700/-) | | |
| E (JUNIORS) (BOBBILI) Starts at 6:45am | Bobbili (Rs.3600/-) | Rambadrapuram (Rs.3450/-) | ■ Arikitota (Rs.34500/-) | Manouram (Rs.3450/-) | Gajapatinagram (Rs.2700/-) | Bondapalli P 5 (Rs.2700/-) | Ambativalasa (Rs.22750/-) | Getiam (Rs.22750/-) | INTU College (Rs.22750/-) | K.L.Puram (Rs.18000/-) | | | | | | |
| F (SENIORS) (SALURU) Starts at 6:45am | Saluru (Rs.3450/-) | ■ BouBommba Rambadrapuram Center (Rs.3600/-) | ■ Marasdum (Rs.3600/-) | Kompatall (Rs.3600/-) | Manapuram (Rs.3600/-) | Gajapatinagram (Rs.2700/-) | Bondapalli P 5 (Rs.2700/-) | Bondapalli (Rs.2700/-) | Ambativalasa (Rs.22750/-) | Gotlam (Rs.22750/-) | RTO Office (Rs.22750/-) | Bridge Down (Rs.18000/-) | | | | |
| G (SENIORS) (NAD - KARASA) Starts at 7:00am | NAD (Rs.3450/-) | Karasu (Rs.3450/-) | ■ Marripalem (Rs.3450/-) | 104 Area (Rs.3450/-) | IT Junction (Rs.3450/-) | Urvasi (Rs.3450/-) | ■ Gnamapuram (Rs.3450/-) | Allipuram (Rs.3450/-) | Railway New Colony (Rs.3450/-) | ■ Donaparthi (Rs.3450/-) | Sangam Office (Rs.3450/-) | Akkayapalem Down (Rs.3450/-) | ■ 4 th Police Station (Rs.3450/-) | PM Palem Stadium (Rs.3450/-) | Boypalem (Rs.3450/-) | |
| H (JUNIORS) (SRIPURAM-VSP) Starts at 7:00am | PMPalemLast Bus Stop (Rs.3450/-) | PMPalem 2nd bus stop, (Rs.3450/-) | ■ PMPalem,1st bus stop, (Rs.3450/-) | Cashed (Rs.3450/-) | Chendramalem (Rs.3450/-) | Madhurawada (Rs.3450/-) | Kommadi (Rs.3450/-) | Rajala Tallvalasa (Rs.25000/-) | Tagarapuvalasa (3 temples) (Rs.25000/-) | Tagarapuvalasa (Rs.25000/-) | Mihilapuri Colony road (Rs.34500/-) | Chandrapalem (Rs.34500/-) | Boypalem (Rs.34500/-) | ■ Beehilli Cross Road (Rs.25000/-) | Tagarapuvalasa (3 temples) (Rs.25000/-) | Avanti College (Rs.25000/-) |
| I (SENIORS) (NAD - NSTI) Starts at 7:00am | NAD (Rs.3450/-) | Baji In (Rs.3450/-) | Gopalapatnam Bunk (Rs.3450/-) | L.G Polymeries (Rs.3450/-) | Vepagunta (Rs.3450/-) | ■ Goshala (Rs.3450/-) | Simhachalam Complex (Rs.3450/-) | Hannumanthawaka (Rs.3450/-) | VIMS (Rs.34500/-) | Yandada (Rs.34500/-) | Midhilapuri Colony road (Rs.34500/-) | Chandrapalem (Rs.34500/-) | Boypalem (Rs.34500/-) | ■ Beehilli Cross Road (Rs.25000/-) | Tagarapuvalasa (3 temples) (Rs.25000/-) | Avanti College (Rs.25000/-) |
| J (JUNIORS) (NAD - NSTI) Starts at 7:00am | NSTL Gate (Rs.3450/-) | ■ Punjab Hotel (Rs.3450/-) | Birla (Rs.3450/-) | ■ Urvasi (Rs.3450/-) | Kancharpalem (Rs.3450/-) | Tatichetlapalem (Rs.3450/-) | Port Hospital Akkayapalem H.W (Rs.3450/-) | ■ Gurudwara (Rs.34500/-) | Maddilapalem (Rs.34500/-) | Automotive (Rs.34500/-) | Isukathota (Rs.34500/-) | Yandada Bunk (Rs.34500/-) | Yendada Bus Stop (Rs.34500/-) | ■ PM palem stadium | Cashed Jatara | |

Fig.8 Fee List page

Application Form

Note : Fresh Application Form

Bus Application Form

Full name: _____

Student id: _____

Contact number: _____

Email: _____

Route: _____

Bus stop: _____

Section: _____

Submit Application

Maharaj Vijayaram Gajapathi Raj College of Engineering (Autonomous)

College Transportation

Bus Feedback Form

Rate: _____

Student id: _____

Section: _____

Bus number: _____

Email: _____

Driver performance: _____

Submit Feedback

© 2024 University Bus Management System
Developed by: M. Koushmatkar & Kartikay Singh Shrestha

Fig.9&10 Application and Feedback page

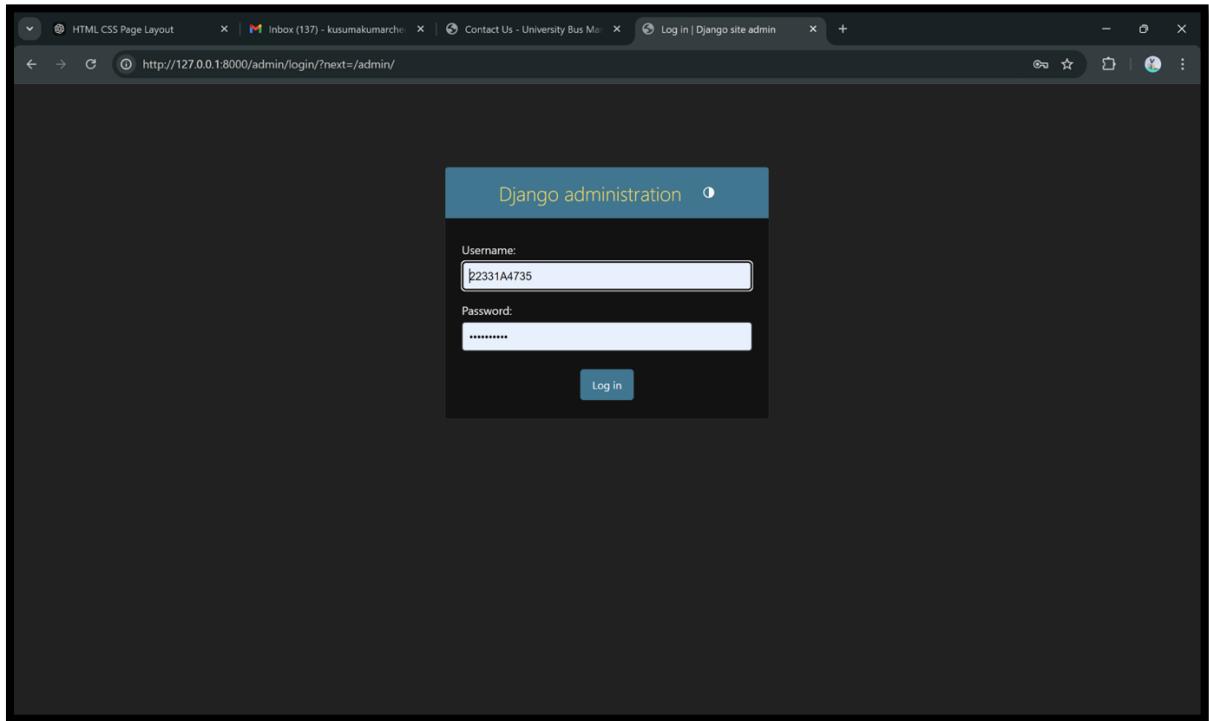


Fig.11 Admin Login page

A screenshot of the MVGR Bus Management System Admin Home page. The title bar says "Site administration | Django site admin". The left sidebar shows "Authentication and Authorization" with "Groups" and "Users" under "kusum". Under "Bus", there are links for "Bus applications", "Bus infos", "Bus namess", "Bus vacancys", "Feedbacks", "Notes", "Student logins", and "Students". The main content area has two sections: "Authentication and Authorization" (Groups and Users) and "Bus" (Bus applications, Bus infos, Bus namess, Bus vacancys, Feedbacks, Notes, Student logins, Students). On the right, a "Recent actions" sidebar lists five entries: "Added 'BUS Z'", "Added 'BUS Y'", "Added 'BUS X'", "Added 'BUS W'", and "Added 'BUS V'".

Fig.12 Admin Home page

The screenshot shows the 'Feedbacks' monitoring page in the Django admin interface. The URL is <http://127.0.0.1:8000/admin/bus/feedback/5/change/>. The left sidebar shows the navigation path: Home > Bus > Feedbacks. The main form contains the following fields:

- Bus ***: BUS A (dropdown menu)
- Student id ***: 22331A4705
- Section ***: CIC
- Buscondition ***: Good
- Email ***: balagakarthikeyanaidu@gmail.com
- Driver performance ***: Good

On the right side of the form, there are five action buttons:

- Save (green)
- Delete (red)
- Save and add another (teal)
- Save and continue editing (blue)
- History (grey)

Fig.13 Feedbacks monitoring page

The screenshot shows the 'Bus vacancies' updation page in the Django admin interface. The URL is <http://127.0.0.1:8000/admin/bus/busvacancy/29/change/>. The left sidebar shows the navigation path: Home > Bus > Bus vacancies. The main form contains the following fields:

- Bus name ***: BUS X
- Total seats ***: 50
- Occupied seats ***: 0

On the right side of the form, there are five action buttons:

- Save (green)
- Delete (red)
- Save and add another (teal)
- Save and continue editing (blue)
- History (grey)

At the bottom of the page, there is a copyright notice: Copyright © 2024 . All rights reserved. and a footer note: Jazzmin version 3.0.1.

Fig.14 Vacancy Updation page

The screenshot shows the Django admin interface for updating a note. On the left is a dark sidebar with navigation links for Django administration, user management, and various bus-related models like Bus applications, Bus infos, Bus names, Bus vacancies, Feedbacks, Notes, Student logins, and Students. The 'Notes' link is currently selected and highlighted in blue. The main content area has two sections: 'Image' and 'Note'. Under 'Image', there's a file input field showing 'media/bus_update.jpg' with a 'Clear' button. Under 'Note', there's a text area containing the text '23/10/24 : J and K Buses Were Not Running.' To the right of the form is a vertical sidebar with five buttons: 'Save' (green), 'Delete' (red), 'Save and add another' (teal), 'Save and continue editing' (blue), and 'History' (grey). At the bottom of the page, there are copyright and version information: 'Copyright © 2024 . All rights reserved.' and 'Jazzmin version 3.0.1'.

Fig.15 Updates Adding page

This screenshot shows the Django admin interface for updating a student record. The sidebar on the left is identical to Fig.15, with the 'Students' link selected. The main form contains several fields: 'Student id *' (22331A4735), 'Full name *' (M.Kusumakumar), 'Course *' (CIC), 'Year *' (3), 'Image' (currently 'media/PHOTO.jpg'), and a series of fee fields: 'First year fee *' (35000.00), 'Second year fee *' (35000.00), 'Third year fee *' (35000.00), 'Fourth year fee *' (0.00), 'Paid amount first year *' (35000.00), 'Paid amount second year *' (35000.00), and 'Paid amount third year *' (10000.00). To the right of the form is a sidebar with five buttons: 'Save' (green), 'Delete' (red), 'Save and add another' (teal), 'Save and continue editing' (blue), and 'History' (grey).

Fig.16 Students Adding page

Database Tables

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database structure with the `bus_busapplication` schema selected.
- Query Editor:** Displays the SQL query: `SELECT * FROM public.bus_busapplication ORDER BY id ASC`.
- Data Output:** Shows the results of the query in a table format.

| | id [PK] bigint | full_name character varying (100) | student_id character varying (20) | contact_number character varying (15) | email character varying (254) | route character varying (50) | bus_stop character varying (100) | created_at timestamp |
|---|-----------------------|--|--|--|--------------------------------------|-------------------------------------|---|-----------------------------|
| 1 | 1 | Kusuma kumar | 22331A4735 | 7207492891 | kusumakumar23@gmail.com | Bus J | Birla Junction | 2024-09- |
| 2 | 2 | Monika | 22331A4754 | 6305876770 | kuuj@gmail.com | Bus D | sklm | 2024-09- |
| 3 | 3 | Chaitanya | 22331A4764 | 7207492891 | jsdbjs@gmail.com | Bus J | Birla Junction | 2024-09- |
| 4 | 4 | Sal Charan | 22331A4710 | 7207492891 | kuuj@gmail.com | Bus J | Birla Junction | 2024-09- |
| 5 | 5 | K Sunitha | 22331A4700 | 7207492891 | jsdbjs@gmail.com | Bus J | Birla Junction | 2024-09- |
| 6 | 6 | K Teja | 22331A4730 | 7207492891 | kusumakumar23@gmail.com | BUS A | Kurmanpalem | 2024-09- |
| 7 | 7 | hari | 12345 | 1234678908 | h@gmail.com | bus d | sklm | 2024-09- |

Total rows: 7 of 7 Query complete 00:00:00.188 Ln 1, Col 1

Fig.17 Applications Table

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database structure with the `bus` schema selected.
- Query Editor:** Displays the SQL query: `SELECT * FROM public.bus ORDER BY id ASC`.
- Data Output:** Shows the results of the query in a table format.

| | id [PK] bigint | name character varying (100) | routes text | morning_start_time time without time zone | college_start_time time without time zone | image character varying (100) | description text |
|---|-----------------------|-------------------------------------|--------------------|--|--|--------------------------------------|------------------------------|
| 1 | 1 | Bus A | (KURMANNAPALEM) | 06:45:00 | 04:10:00 | bus_images/mvgrbus.jpg | Good Morning Students |
| 2 | 2 | Bus B | Scindia | 06:45:00 | 04:00:00 | bus_images/mvgrbus_pRaRTKl.jpg | hello |
| 3 | 3 | Bus C | Srikakulam | 06:45:00 | 04:00:00 | bus_images/mvgrbus_r8GbCMV.jpg | Hello Good Morning |
| 4 | 4 | Bus D | Srikakulam | 06:45:00 | 04:00:00 | bus_images/mvgrbus_LZUXQqs.jpg | Your Bus Timings are changed |
| 5 | 5 | Bus E | Bobbili | 06:45:00 | 04:00:00 | bus_images/mvgrbus_gBUODUo.jpg | Hello Students |

Total rows: 5 of 5 Query complete 00:00:00.221 Ln 1, Col 1

✓ Successfully run. Total query runtime: 221 msec. 5 rows affected. X

Fig.18 Bus Routes Table

Conclusion

The University Bus Management System provides an efficient, streamlined, and user-friendly solution for managing university transportation services. This web-based platform, developed using Django, meets all the specified requirements, including real-time seat availability, route management, fee structure display, and service feedback. By automating key processes like registration, seat reservation, and route information updates, the system reduces administrative workload and enhances accessibility for students and staff alike.

Through the integration of PostgreSQL for secure and scalable data management, the system ensures reliable storage and retrieval of user and transportation data. The inclusion of RESTful APIs supports seamless communication between front-end and back-end, creating a responsive and engaging experience for end-users. Additionally, the system's ability to handle feedback and provide continuous updates empowers administrators to improve service quality proactively.

In conclusion, the proposed University Bus Management System achieves the project's objectives by offering a comprehensive, adaptable, and efficient platform for managing transportation logistics. This solution not only enhances operational efficiency but also contributes to a better experience for students, aligning with modern technological standards and ensuring long-term scalability and usability.

Future Scope and Developments

Mobile Application Development: A mobile app for students, faculty, and administrators would significantly increase accessibility and ease of use. Features like real-time bus tracking, notifications for delays, and seat availability could be integrated.

Real-Time Bus Tracking: Incorporating GPS tracking for buses would allow students and faculty to see the real-time location of their bus, estimated arrival times, and delays.

Automated Payments: Adding an online payment system for transportation fees can streamline the payment process for students and faculty. Integration with payment gateways will allow users to pay transport fees directly through the system.

Automated Notifications: The system could be enhanced to send automated SMS or email notifications for bus delays, route changes, or any other important updates related to transport services

Advanced Reporting and Analytics: The system could provide advanced analytics and reporting tools, allowing the administration to analyse route efficiency, student usage, and cost management. This could help optimize routes and reduce costs.

Integration with Other Campus Systems: Integration with other campus management systems (such as student information systems) could create a more unified experience. This integration could synchronize data between systems and avoid duplicating information.

User Feedback and Complaints Handling: A more sophisticated module for feedback and complaints handling could be developed, including tracking the status of complaints and allowing users to rate the services.

Cloud-Based Solution: Migrating the system to a cloud-based infrastructure would enhance scalability, security, and accessibility, allowing more users to access the system simultaneously and providing flexibility for future expansions.

Machine Learning for Predictive Maintenance: Using machine learning algorithms, the system could predict maintenance needs for buses, reducing downtime and improving reliability.

Multi-Language Support: To accommodate a more diverse user base, the system could offer multi-language support, ensuring ease of use for all students, staff, and faculty members.

References

BOOKS:-

- “Introduction to Cloud Computing Architecture” 1st Edition June 2009, Sun Microsystems Inc.
- Pankaj Jalote, “An approach to software engineering”, third edition, 2005, Narosa Publishing House.
- Leon & Leon, “Database Management System”, Vikas Publishing House. Elmasri , Navathe,” Fundamentals of database systems ”,addition Wesley

WEBSITES:-

1. www.google.co.in
2. www.wikipedia.com
3. www.slideshare.net
4. www.webpoint.com
- 5.