TRANSPORT MANAGEMENT SYSTEM

A Major Project Report Submitted in Partial Fulfillment

for the Award of the Degree of

Bachelor of Technology in Computer Science and Engineering

To

Dr. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY,

LUCKNOW

Submitted by:

AASHI SRIVASTAVA (1900100100001)

KRITI SRIVASTAVA (1900100100079)

MANAS DUBEY (1900100100081)

SARTHAK SHARMA (1900100100128)

Under the supervision of

MR. DHARMENDRA KUMAR

ASSOCIATE PROFESSOR

UNITED
Group of Institutions
Greater Noida • Allahabad

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNITED COLLEGE OF ENGINEERING AND RESEARCH,

PRAYAGRAJ

MAY 2023

TRANSPORT MANAGEMENT SYSTEM

A Major Project Report Submitted in Partial Fulfillment

for the Award of the Degree of

Bachelor of Technology in Computer Science and Engineering

To

Dr. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY,

LUCKNOW

Submitted by:

AASHI SRIVASTAVA (1900100100001)

KRITI SRIVASTAVA (1900100100079)

MANAS DUBEY (1900100100081)

SARTHAK SHARMA (1900100100128)

Under the supervision of

MR. DHARMENDRA KUMAR

ASSOCIATE PROFESSOR

UNITED
Group of Institutions
Greater Noida • Allahabad

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNITED COLLEGE OF ENGINEERING AND RESEARCH,

PRAYAGRAJ

MAY 2023

CANDIDATE'S DECLARATION

We, hereby certify that the project entitled "**Transport Management System**" submitted by us in partial fulfillment ofthe requirement for the award of degree of the B. Tech. (Computer Science & Engineering) submitted to Dr. A.P.J. Abdul Kalam Technical University, Lucknow at United College of Engineering and Research, Prayagraj is an authentic record of our own work carried out duringa period from June, 2022 to May, 2023 under the guidance of Prof.---(Mr. Dharmendra Kumar, Assistant Professor, Department of Computer Science & Engineering). The matter presented in this project has not formed the basis for the award of anyother degree, diploma, fellowship or any other similar titles.

(AASHI SRIVASTAVA, 1900100100001)

(KRITI SRIVASTAVA, 19001001000079)

(MANAS DUBEY, 1900100100081)

(SARTHAK SHARMA, 1900100100128)

Place:

Date:

CERTIFICATE

This is to certify that the project titled "**Transport Management System**" is the bona fide work carried out by AASHI SRIVASTAVA (1900100100001), KRITI SRIVASTAVA (1900100100079), MANAS DUBEY (1900100100081) & SARTHAK SHARMA (1900100100128) in partial fulfillment of the requirement for the award of degree of the B. Tech. (Computer Science & Engineering) submitted to Dr. A.P.J Abdul Kalam Technical University, Lucknow at United College of Engineering and Research, Prayagraj is an authentic record of their own work carried out duringa period from June, 2022 to May, 2023 under the guidance of Prof.- (Mr. Dharmendra Kumar, Assistant Professor, Department of Computer Science & Engineering). The Major Project Viva-Voce Examination has been held on _____.

Signature of the Guide _____

[Mr. Dharmendra Kumar]

Signature of Project Coordinator _____

[Mr. Shyam Bahadur Verma]

Signature of the Head of Department _____

[Dr. Vijay Kumar Dwivedi]

Place:

Date:

ABSTRACT

The Transportation Management System (TMS) is an innovative and comprehensive solution designed to streamline and improve campus transportation services. As more students and staff rely on university transportation, the need for an efficient and organized system becomes essential. TMS provides a digital platform that enables management, monitoring and optimization of a wide range of transport activities, leading to increased efficiency, comfort and safety.

TMS incorporates advanced technologies such as GPS tracking of assigned bus, web applications and allotment of bus pass to provide real-time information on vehicle locations, routes and schedules. This gives students and faculty access to accurate and up-to-date information on available transportation, estimated arrival times, and timetable changes. Through an easy-to-use web application, users can easily view bus passes, track assigned vehicles, and receive notifications of delays or cancellations or notices.

The system will also benefit college administrators and transportation managers by providing a centralized platform to efficiently manage the entire fleet of transportation vehicles. Easily assign vehicles and drivers, create optimized routes, and monitor vehicle performance and maintenance schedules. Integrating data analytics gives managers valuable insight into transportation patterns, demand variability and user preferences, facilitating informed decision making and resource allocation.

Deploying the Transportation Management System provides many benefits, including increased operational efficiency, reduced wait times, enhanced security measures, and increased user satisfaction. By using this digital solution, colleges can optimize transport services, making them more reliable and convenient.

ACKNOWLEDGEMENT

We express our sincere gratitude to the Dr. A.P.J Abdul Kalam Technical University, Lucknow for giving us the opportunity to work on the Major Project during our final year of B.Tech. (CSE) is an important aspect in the field of engineering.

We would like to thank Dr. H.P. Shukla, Principal and Dr. Vijay Kumar Dwivedi, Head of Department, CSE at United College of Engineering and Research, Prayagraj for their kind support.

We also owe our sincerest gratitude towards Prof. **Dharmendra Kumar** for his valuable advice and healthy criticism throughout our project which helped us immensely to complete our work successfully.

We would also like to thank everyone who has knowingly and unknowingly helped us throughout our work. Last but not the least, a word of thanks for the authors of all those books and papers whichwe have consulted during our project work as well as for preparing the report.

Table of Contents

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Motivation and Methodology

The transport service of our college is one of the most beneficial and important facilities provided by the college for the day scholars and the staff residing in different parts of the city for their day to day up-down.
Transportation plays an important role in the efficient functioning of campuses, providing convenience and accessibility to students, faculty and staff. As the university's reliance on transportation services grows, so does the need for a well-organized system to manage and optimize these services.

The Transport Management System (TMS) includes various functions such as vehicle allocation, driver allocation, route planning, schedule management, real-time information distribution, etc. TMS aims to improve efficiency, accessibility, safety and user satisfaction by integrating GPS tracking, and web applications.

Providing real-time information is a key feature of TMS. Students and faculty can access up-to-date information on transportation options, estimated arrival times, and timetable changes through an easy-to-use web application. This allows you to plan your trips effectively, minimize wait times and make informed decisions about your transportation needs.

Additionally, TMS aims to improve the overall user experience by providing an easy-to-use web application. Students and faculty can easily track assigned vehicles, and receive timely notifications of delays and changes. Effective communication channels are established between drivers, students and staff to ensure smooth operations and prompt resolution of concerns.

With the growing focus on environmental sustainability, TMS promotes environmentally friendly transportation practices on campus. By optimizing routes, reducing fuel consumption and minimizing emissions, the system contributes to a greener and more sustainable transportation environment.

1.2 Objectives

- Create a System of Transport Management: The main objective of TMS is to optimize transportation management in universities. By automating and simplifying tasks, administrators can efficiently allocate vehicles and drivers, create optimized routes, and manage schedules. This streamlines processes, reduces manual effort, and makes efficient use of resources.

- Accessibility of real-time information: The purpose of this system is to provide students and faculty with easy access to accurate and up-to-date information on available transportation, estimated arrival times, and timetable changes. This improves user convenience and reduces waiting time.

- Improved User Experience: The system aims to improve the overall user experience by providing an easy-to-use mobile application for booking rides, tracking vehicles, and receiving notifications. It also enables effective communication between drivers, students and staff, ensuring smooth operations and prompt response to user concerns.

- Efficient transportation management: TMS aims to streamline various transportation activities such as vehicle allocation, driver allocation, route planning and schedule management. The aim is to optimize the use of resources and ensure smooth operation of transport vehicles.

1.3 Scope

- Automation of Bus Allocation: Automating bus allocation with the Transport Management System offers benefits such as intelligent scheduling, real-time demand response, route optimization, integrated vehicle tracking, resource management. These advances will optimize efficiency, reduce congestion, and improve the user experience of university transportation.

- Automatic Route Optimization: Automation can enable TMS to optimize bus routes based on various factors such as traffic conditions, passenger locations, and time constraints. The system can continuously analyze data and make real-time adjustments to bus routes, ensuring efficient transportation and minimizing travel time for students and staff.

- Real-time Demand-Responsive Allocation: TMS can integrate real-time data sources such as student and staff inputs, mobile application, and sensors within buses to dynamically allocate buses based on demand. By monitoring passenger flow and demand fluctuations, the system can automatically adjust the allocation of buses to ensure optimal utilization and avoid overcrowding or underutilization of buses.

1.4 Existing System

The pen-and-paper method of managing transportation services may be familiar and easy, but it has some notable drawbacks when compared to digital systems like the Transport Management System (TMS). The pen and paper method has the following drawbacks:

- Manual and time-consuming: Pen-and-paper methods require manual entry and management of data, which can be time consuming and prone to human error. Tasks such as vehicle allocation, driver allocation, and route planning must be performed manually, leading to inefficiencies and delays.

- Limited Accessibility and Availability: Pen-and-paper records are physical documents and are not readily accessible or available to all parties. Information about transportation options, timetables, or changes may not be easily disseminated, resulting in communication gaps and user confusion.

- Lack of real-time updates: Pen-and-paper methods may not immediately communicate or record updates to transportation schedules or changes in vehicle availability. This lack of real-time updates can lead to outdated information, inconvenience and inefficient transportation planning.

- Data organization and storage issues: Storing and organizing transportation data on physical documents can be difficult. Retrieving specific information or tracking historical records can be labor intensive and not as efficient as structured databases and digital systems that provide search capabilities.

- Security and Data Loss Risk: Physical documents are vulnerable to damage, loss, or unauthorized access. In the event of fire, theft or loss, transportation records can be lost, disrupting business and making recovery of critical information difficult.

- Lack of integration and communication: Pen-and-paper methods may not integrate seamlessly with other systems and platforms. Communication between drivers, administrators and users can become inefficient, resulting in delays and miscommunications.

In summary, pen-and-paper methods may be familiar, but they have limitations such as manual work, limited access, lack of real-time updates, data confusion, and security risks. The introduction of digital systems like TMS overcomes these shortcomings and offers significant advantages in terms of efficiency, accuracy, real-time information, data analysis and improved communication.

CHAPTER 2

FEASIBILITY STUDY

A feasibility study is an analysis and evaluation of a proposed project or business idea to determine its viability and potential success. It typically involves a detailed examination of the various aspects of the project, including its technical, economic, legal, and operational aspects.

The purpose of a feasibility study is to assess whether the proposed project is feasible, practical, and economically viable. The study helps identify potential risks, challenges, and obstacles that may impact the project's success. It also helps to determine the resources, time, and effort required to complete the project and achieve its goal.

2.1 Technical Feasibility

Technical feasibility is an assessment of whether a proposed system or solution can be successfully implemented using available technology and resources. For the Transport Management System (TMS), it is important to assess technical feasibility prior to deployment.

The technical feasibility of implementing a Traffic Management System (TMS) using AngularJS, ASP.NET, and MySQL can be assessed based on the following considerations.

1. **AngularJS:** AngularJS is a JavaScript framework for building web applications. It provides a structured approach to developing dynamic and responsive user interfaces. The technical feasibility of using AngularJS for TMS depends on having qualified developers familiar with AngularJS development. Additionally, you should assess the compatibility of his AngularJS with other components of your system, such as backends and databases, to ensure smooth integration.

2. **ASP.NET:** ASP.NET is a widely used framework for building web applications using the .NET platform. It provides powerful tools and libraries for developing scalable and secure applications. The technical feasibility of using ASP.NET for TMS depends on the availability of developers with ASP.NET development experience. Compatibility with other components such as front-end frameworks (AngularJS) and databases (MySQL) must be considered to ensure seamless integration.

3. **MySQL:** MySQL is a popular open-source relational database management system. Provides robust data storage, retrieval, and management capabilities. The technical feasibility of using MySQL for TMS depends on the compatibility of the chosen development framework (ASP.NET) with MySQL and its ability to handle the expected data volume and performance requirements. To ensure proper implementation and optimization, it is important to assess your existing expertise in MySQL database administration and maintenance.

4. **Integration:** The technical feasibility of integrating AngularJS, ASP.NET, and MySQL depends on their compatibility and ability to communicate seamlessly. To establish smooth communication between frontend (AngularJS), backend (ASP.NET) and database (MySQL), you need to implement APIs and data exchange protocols. A feasibility assessment should consider the availability of libraries, frameworks, and resources for integration.

5. **Scalability and Performance:** The technical feasibility of using this technology stack for TMS must also consider scalability and performance requirements. Systems must be able to handle increasing numbers of users, vehicles and data without sacrificing performance. Load testing and optimization techniques should be employed to ensure that the system meets the expected performance.

6. **Development and Maintenance Resources:** The technical feasibility of using AngularJS, ASP.NET, and MySQL for TMS also depends on the availability of development and maintenance resources. Qualified developers with knowledge of these technologies should be available for system development, updates and ongoing maintenance.

Evaluating technical factors such as developer proficiency, compatibility, integration skills, scalability, and resource availability helped determine the feasibility of implementing her TMS using AngularJS, ASP.NET, and MySQL. increase.

2.2 Economic Feasibility

The feasibility study aims to evaluate the economic viability of the project by comparing the costs and benefits. The cost-benefit analysis serves as the basis for assessing the economic feasibility. While the new system requires additional initial investment compared to the existing method, it is justified by the anticipated improvement in service quality. The study should focus on areas such as accuracy, timeliness, cost comparison, estimating the hardware's lifespan, and overall project objectives.

2.3 Legal Feasibility

Since this project needs no copyrighting, patenting, and doesn't have any relation with anybody else's property rights, it can be considered as a legally feasible project.

2.4 Time Feasibility

As it has been more probable (as per the requirements and functions specifications of the system) that the project can be completed within the given timeframe, it is considered that undertaking this project is feasible in the context of time.

2.5 Social Feasibility

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change.

The new proposed system is very much useful to the users and Social feasibility is determination of whether a proposed project will be acceptable to people or not, So this project is totally Social and Feasible .

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 Software Requirements

Description	Type
Operating System	**Windows XP / Windows**
Programming Languages	**ASP.NET Core MVC, HTML, JavaScript**
Frontend Framework	**AngularJS**
Database	**MySQL**
IDE	**Visual Studio**
Browser	**Google Chrome**
Server	**XAMPP**

3.2 Hardware Requirements

Description	Type
System	**v Pentium 4, Intel Core i3, i5, i7 and 2 GHz minimum. OR Android 6.0 or above with GPS technology**
RAM	**2 GB or above**
Hard Disk	**10 GB or above**
Input/Output	**Keyboard, Mouse / Monitor**

3.3 Functional Requirements

Functional requirements describe specific functions, capabilities, and actions that a system, such as a transportation management system (TMS), must perform to meet user needs. Here are some examples of TMS functional requirements.

1. **User Registration and Authentication:** Systems must provide user registration functionality to allow students, faculty and staff to create accounts and authenticate their identity to access the TMS.

2. **Vehicle and driver management:** TMS should allow administrators to manage vehicle fleets and assign drivers to specific routes and schedules. This includes adding new vehicles, updating vehicle details, managing driver information, and tracking driver availability.

3. **Route planning and optimization:** The system aims to enable efficient route planning, taking into account factors such as distance, traffic conditions and user demand. The aim is to optimize routes to minimize travel times, fuel consumption and operating costs.

4. **Schedule management:** TMS should allow administrators to create and manage transportation schedules, including defining pick-up and drop-off locations, time slots, and service frequency. It should support dynamic planning and allow adjustments due to changing demand and unforeseen circumstances.

5. **Real-time tracking and monitoring:** The system should provide real-time tracking and monitoring of vehicles, allowing users to see the current location of assigned vehicles and estimate arrival times. This feature increases user comfort and reduces waiting time.

6. **Notifications and Alerts:** The system should send users timely notifications and alerts of schedule changes, delays, or cancellations. This keeps users informed of any updates or glitches in the transport service.

3.4 Non-Functional Requirements

Non-functional requirements are the qualities and constraints that describe how the Transportation Management System (TMS) should perform and behave, rather than specific features. Here are some examples:

- **Speed:** The TMS should respond quickly to user actions and perform tasks without significant delays, so you don't have to wait long for the system to work.

- **Reliability:** The TMS should be dependable and available whenever you need it, without frequent errors or disruptions, so you can rely on it for your transportation needs.

- **Scalability:** The system should be able to handle a growing number of users and transportation demands without slowing down or crashing, so it can accommodate more people as needed.

- **Security**: The TMS should keep your personal information and data safe from unauthorized access, ensuring your privacy and protecting you from potential risks.

- **User-Friendliness:** The system should be easy to use and navigate, with clear instructions and feedback, so you can quickly understand how to use it without confusion.

- **Compatibility:** The system should work well with different devices and web browsers, making it convenient for you to access and use the TMS from your preferred device.

- **Maintainability**: The system should be designed in a way that makes it easy to fix bugs, update features, and keep it running smoothly for a long time without causing unnecessary difficulties.

CHAPTER 4

DESIGN AND PLANNING

4.1 Software Development Life Cycle Model

4.1.1 Iterative Waterfall - Model

The Iterative Waterfall model is an extension of the traditional Waterfall model, which is a linear and sequential approach to software development. The Iterative Waterfall model introduces iterations or cycles within each phase of the Waterfall model, allowing for feedback, adjustments, and refinements throughout the development process.

In the Iterative Waterfall model, the software development process is divided into phases, similar to the Waterfall model. These phases typically include requirements gathering, system design, implementation, testing, deployment, and maintenance. However, unlike the strict sequential nature of the Waterfall model, the Iterative Waterfall model allows for the repetition of these phases in cycles.

4.1.2 Phases of Iterative Waterfall Model

Here are the key characteristics and steps of the Iterative Waterfall model:

1. **Requirement Gathering:** In this phase, the requirements for the software are collected and documented. This includes gathering information from stakeholders, analysing user needs, and defining the scope of the project.

2. **System Design:** The gathered requirements are translated into a system design. This phase involves creating architectural diagrams, high-level designs, and defining the overall structure of the software.

3. **Implementation:** The implementation phase focuses on coding and developing the software based on the system design. The coding work is divided into modules or components that are developed and integrated.

4. **Testing**: Once the implementation is complete, testing activities are performed to ensure that the software meets the specified requirements. Different types of testing, such as unit testing, integration testing, system testing, and acceptance testing, are conducted to verify the functionality and quality of the software.

5. **Deployment:** After successful testing and approval, the software is deployed or released to the end-users or clients. This phase involves installation, configuration, and any necessary data migration tasks.

6. **Maintenance:** The maintenance phase involves ongoing support and enhancements for the deployed software. It includes addressing bug fixes, incorporating user feedback, and making necessary updates and improvements.

In the Iterative Waterfall model, after completing one cycle of these phases, a review or evaluation is conducted. The feedback obtained from the evaluation is then used to refine and improve the software in subsequent iterations or cycles. Each iteration builds upon the previous one, incorporating changes based on the feedback received.

The key advantage of the Iterative Waterfall model is its ability to accommodate changes and feedback within the sequential structure of the Waterfall model. It allows for iterative refinements and adjustments, reducing the risks associated with a linear approach. However, it's important to note that the Iterative Waterfall model may not provide the same level of flexibility as more iterative and incremental models like the Spiral or Agile methodologies.

Overall, the Iterative Waterfall model is suitable for projects that require a structured and systematic approach, yet allow for some level of feedback and adaptation throughout the development process.

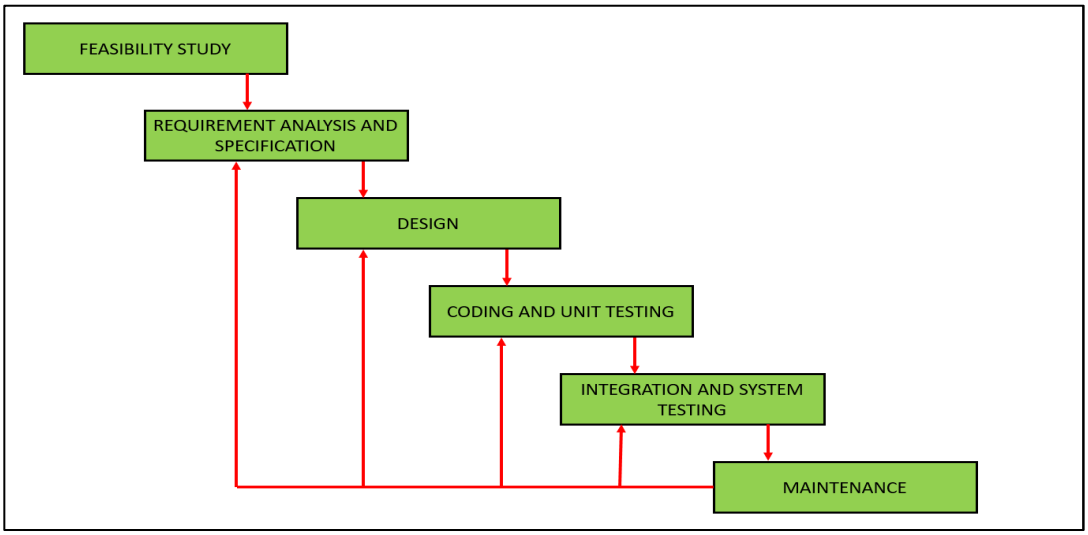


Fig 4.1 Iterative Waterfall Model

4.1.3 Pros and Cons

The Iterative Waterfall model offers both advantages and disadvantages. Here are the pros and cons of using this approach:

Pros:

1. **Structured Approach:** The model provides a structured and systematic framework for software development. It follows a well-defined sequence of phases, which can be beneficial for projects that require a clear plan and a disciplined approach.

2. **Early Feedback and Adjustments:** The iterative nature of the model allows for feedback and adjustments throughout the development process. This enables early detection of issues or shortcomings, leading to prompt corrective actions and reducing the risk of major problems later on.

3. **Reduced Risk of Scope Creep:** By dividing the development process into iterations, the model helps manage scope creep. Each iteration focuses on a subset of requirements, allowing for better control over project scope and avoiding overwhelming changes or additions that can derail the project.

4. **Documentation and Traceability:** The model encourages comprehensive documentation at each phase, making it easier to trace decisions, requirements, and design choices. This documentation can be valuable for future reference, maintenance, and knowledge transfer.

5. **Predictability and Project Planning:** The iterative nature of the model, combined with the structured approach, provides a level of predictability for project planning. The timeline, milestones, and deliverables can be estimated with greater accuracy compared to more flexible or adaptive methodologies.

Cons:

1. **Limited Flexibility:** While the model allows for iterative feedback and adjustments, it may not offer the same level of flexibility as more agile methodologies. Once a phase is completed and the project moves forward, it can be challenging to incorporate significant changes without affecting the overall timeline and process.

2. **Late Discovery of Major Issues:** Despite the iterative approach, major issues or requirements gaps may only surface during later iterations or even in the deployment phase. This can result in rework, delayed timelines, and increased costs.

3. **Difficulty in Managing Dependencies:** The sequential nature of the model can lead to challenges in managing dependencies between different phases or components. Changes or delays in one phase may have a cascading effect on subsequent phases, making it necessary to carefully coordinate activities and dependencies.

4. **Limited User Involvement:** The model primarily focuses on feedback from stakeholders and clients rather than extensive user involvement. This can result in a potential disconnect between the final product and the actual needs and expectations of end-users.

5. **Difficulty in Handling Changing Requirements:** While the model can accommodate some level of changes through iterations, significant changes or shifting requirements may disrupt the sequential nature of the process. This can lead to increased complexity, rework, and potential delays.

It's important to consider these pros and cons in the context of your specific project requirements, team capabilities, and organizational environment. The Iterative Waterfall model may be suitable for projects that benefit from a structured approach with some flexibility for feedback and adjustments but may not be the best fit for projects requiring high adaptability or rapid changes in requirements.

4.2 Database

4.2.1 Entities Details

Entity	Description
BID	Stores Bus ID of the Buses
BusNumber	Stores Bus Number of Buses
VisitingAreas	Stores all the visiting areas of the Bus
DriverName	Stores Name of the Driver
DriverMobile	Stores Mobile Number of Driver
LID	Stores Location ID
Longitude	Stores Longitude of Location of Bus
Latitude	Stores Latitude of Location of Bus
LastUpdateTime	Stores the Last Updated Time of Bus
NID	Stores Notice ID
Title	Stores Title of Notice
Description	Stores Description of Notice
UpdateTime	Stores Time at which Notice was Updated
SID	Stores Student ID
StudentName	Stores Student Name
Email	Stores Email of Student
Mobile	Stores Mobile Number of Student
Address	Stores Address of Student

Password	Stores Password of Student
UID	Stores User ID
UserName	Stores User Name
Email	Stores Email of Admin
Password	Stores password of Admin
Mobile	Stores Mobile Number of Admin
UserType	Stores Type of User-Admin / Student
isVerified	Checks whether the user is verified from college
Status	Stores Status of Bus
DOB	Stores Date of birth of users

4.2.2 ER Diagram

The ER diagram for a College Transport Management System typically includes the following entities, their attributes, and relationships:

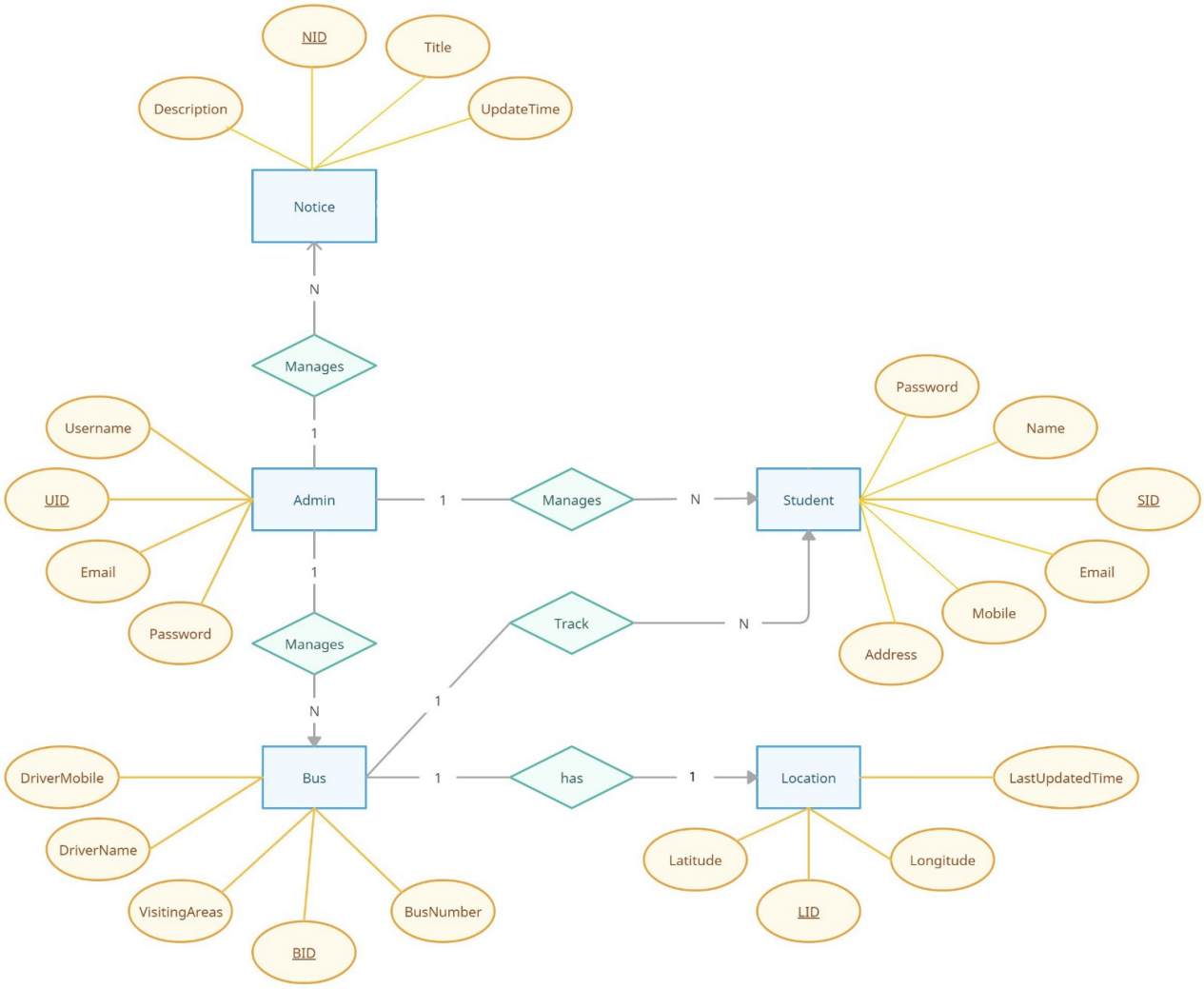
1. Entity: Admin
    - Attributes: User ID, Email, Password, UserType, Email

2. Entity: Bus
    - Attributes: Bus ID, Bus Number, Visiting Areas, Driver Name, Driver Mobile

3. Entity: Student
    - Attributes: Student ID, Name, Address, Phone, Email, Password, Bus ID

4. Entity: Notice
    - Attributes: Notice ID, Title, Description, Updated Time

5. Entity: Location
    - Attributes: Location ID, Longitude, Latitude, Bus ID, Last Updated Time


NID
Title
Description
UpdateTime
Notice
N
Manages
1
Username
UID
Email
Password
Admin
1
Manages
N
Student
Password
Name
SID
Email
Mobile
Address
1
Manages
N
Track
N
1
DriverMobile
DriverName
VisitingAreas
BID
BusNumber
Bus
1
has
1
Location
LastUpdatedTime
Latitude
LID
Longitude

ER-Diagram

4.2.3 Use Case Diagram

A use case diagram provides a high-level visualization of the interactions between actors (users or external systems) and the system itself. In the context of a college transport management system, here's an explanation of the key actors and their interactions as depicted in a use case diagram:

Student: The student is the primary actor in the system as they interact directly with the transport management system to avail transportation services. The use cases involving the student actor may include:

View Schedule: The student can view the available schedules for different routes.

View Bus Pass: The student can see their Bus Pass for their assigned bus.

Administrator: The administrator is responsible for managing and maintaining the transport management system. They perform administrative tasks and oversee system operations. Use cases involving the administrator may include:

Add/Edit Routes: The administrator can add new routes or modify existing route information.

Manage Vehicles: The administrator can add, edit, or remove vehicles from the system.

Assign Drivers: The administrator can assign drivers to specific routes and schedules.

Driver: The driver is responsible for operating the college transport vehicles and ensuring the safe transportation of students. Use cases involving the driver may include:

View Assigned Schedule: The driver can view their assigned schedule for the day.

Update Status: The driver can update the status of their route (e.g., on time, delayed).

Communicate with Administrator: The driver can communicate with the administrator for route updates or emergencies.

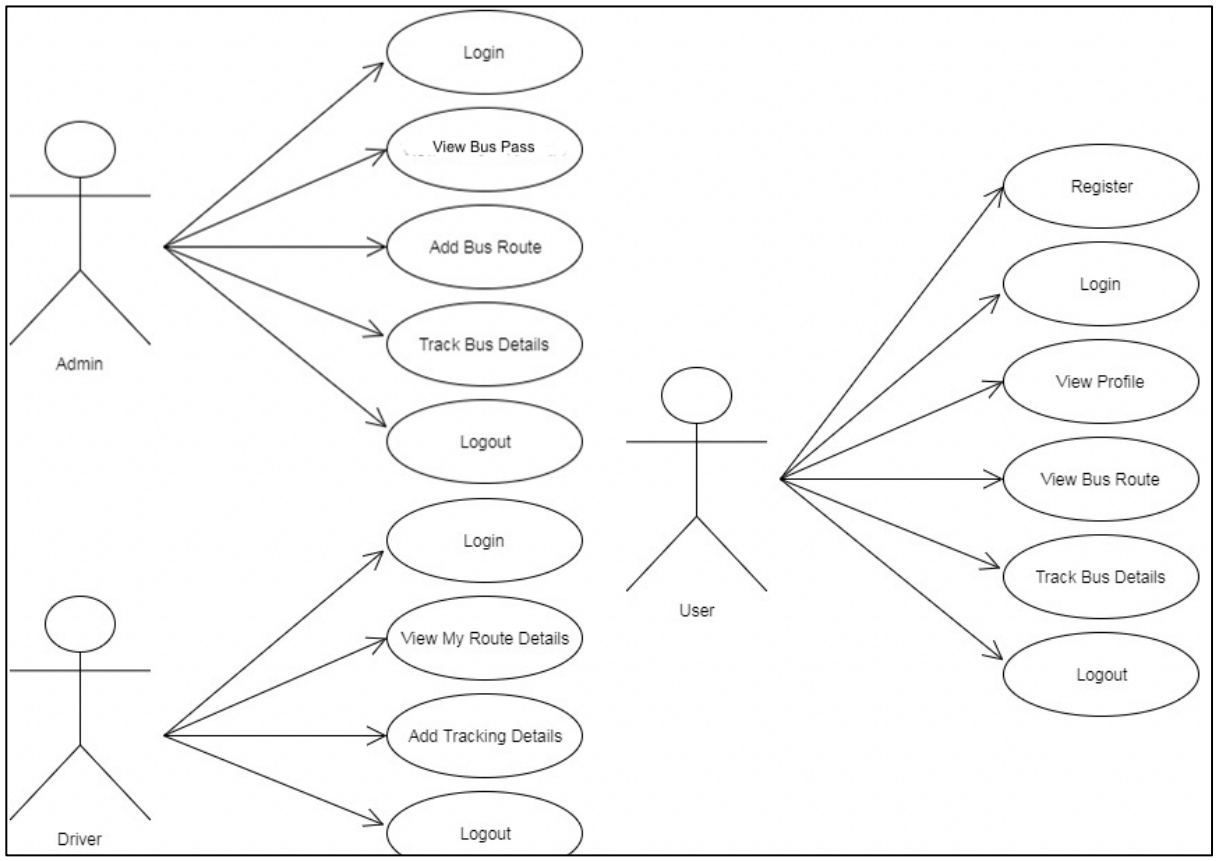


Fig 4.3 Use Case Diagram

4.2.4 Class Diagram

A class diagram is a visual representation of the classes, relationships, attributes, and methods in an object-oriented system. In the context of a college transport management system, here's an explanation of the key classes and their relationships as depicted in a class diagram:

Student class:

Attributes: student_id, name, contact_details, ...

Registered Bus class:

Attributes: Bus_id, student_id, route_id, schedule_id,...
Associations: Registered bus class has associations with Student class (many-to-one association), Route class (many-to-one association), and Schedule class (many-to-one association).

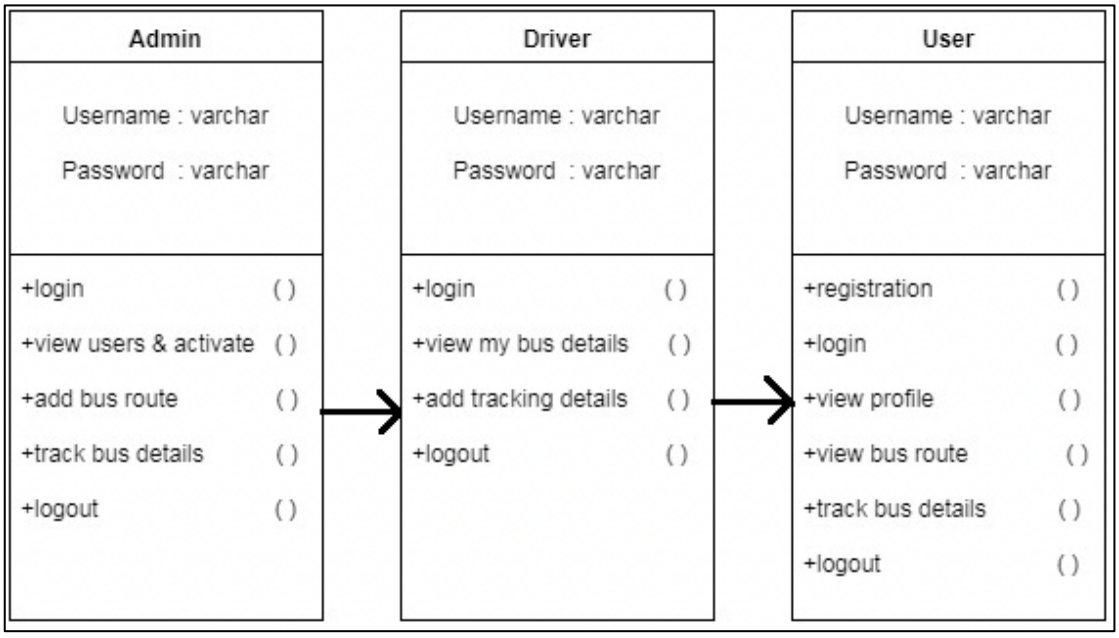


Fig 4.4 Class Diagram

Route class:

Attributes: route_id, starting_point, endpoint, distance, ...
Associations: Route class can have a one-to-many association with Schedule class (one route can have multiple schedules).

Schedule class:

Attributes: schedule_id, route_id, departure_time, arrival_time, ...
Associations: Schedule class has associations with Route class (one-to-many association) and Booking class (one-to-many association).

Vehicle class:

Attributes: vehicle_id, make, model, capacity, ...
Associations: Vehicle class can have a one-to-many association with Route class (one vehicle can be assigned to multiple routes).
Driver class:

Attributes: driver_id, name, contact_details, ...
Associations: Driver class can have a one-to-many association with Route class (one driver can be assigned to multiple routes).

These classes represent the major entities involved in the college transport management system. The associations between classes depict the relationships between them, such as how they are related and connected. For example, a Route can have multiple Schedules, and a Vehicle and Driver can be associated with multiple Routes.

4.2.5 Activity Diagram

An activity diagram is a graphical representation of the flow of activities or processes within a system. In the context of a college transport management system, here's an explanation of the key activities and their flow as depicted in an activity diagram:

1. **Administrator Management Process:**

    - The activity starts with the administrator accessing the system.
    - The administrator can perform various management tasks, such as adding or modifying routes, schedules, vehicles, and drivers.
    - Each management task is represented as a separate activity within the diagram.
    - After completing a management task, the administrator can choose to continue with another task or exit the system.
    - The activity diagram ends when the administrator exits the system.

2. **Driver Operations:**

    - The activity starts with the driver accessing the system.
    - The driver views their assigned schedule and route.
    - The driver updates the status of their route, indicating if they are on time, delayed, or have encountered an incident.
    - The driver can communicate with the administrator for any updates or emergencies.
    - After completing their operations, the driver can choose to continue with other tasks or exit the system.
    - The activity diagram ends when the driver exits the system.

3. **System Reporting:**

    - The activity starts with the generation of reports by the system.

- The system retrieves relevant data, such as booking records, route information, or driver assignments.
- The system processes and analyses the data to generate reports.
- The generated reports can include information on booking statistics, route utilization, or driver performance.
- The activity diagram ends with the presentation or distribution of the generated reports.

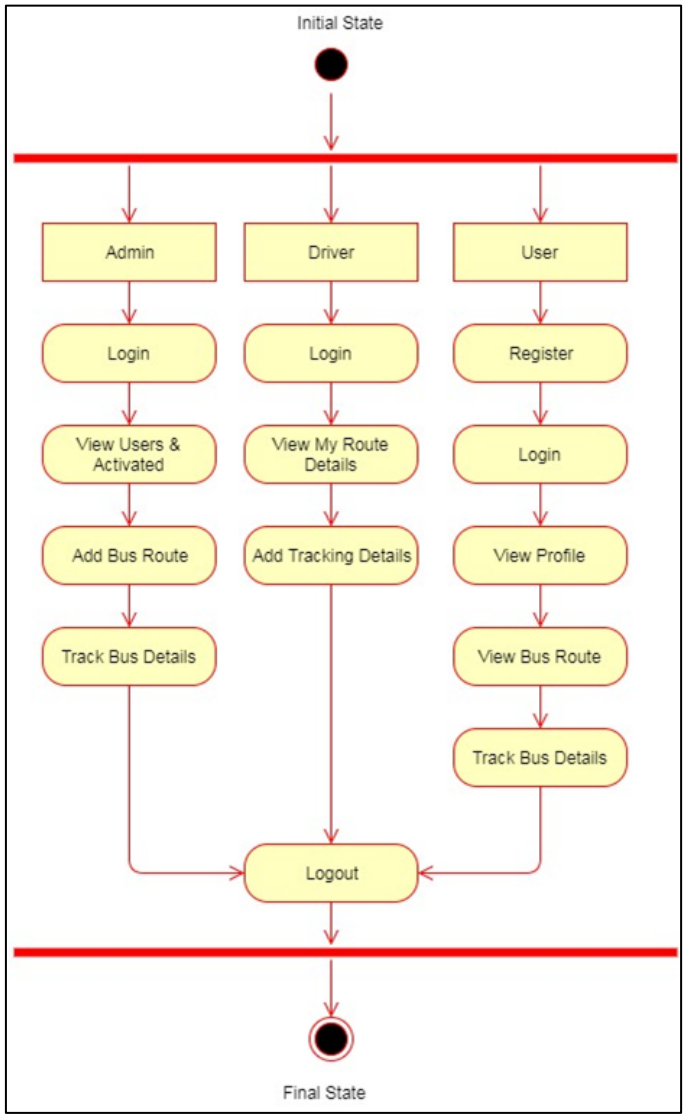


Fig 4.4 Activity Diagram

4.3 Description of the data types managed by the system

4.3.1 Description of the data types managed by the system

The College Transport Management System manages various data types related to different entities and their attributes. Here's a description of the commonly managed data types in the system:

Text/String: Used to store textual information such as names, addresses, phone numbers, and email addresses of entities like colleges, departments, students, drivers, and routes.

Numeric/Integer: Used to store numerical data such as unique identifiers (IDs) for entities like colleges, departments, students, vehicles, drivers, routes, transport schedules, and bookings. It can also be used to represent capacities, license numbers, and contact numbers.

Date/Time: Used to store dates and times for attributes such as the booking date, departure time, arrival time, and schedule dates.

Boolean: Used to store binary or true/false values. For example, the "Status" attribute in the Booking entity can indicate whether a booking is confirmed or not.

Enumerations/Enums: Used to define a predefined list of values for attributes. For example, the Department entity may have an attribute called "Head of Department" that can have values like "Professor," "Associate Professor," "Assistant Professor," etc.

Floating-Point/Decimal: Used to store numeric values with decimal places. This data type may be used for attributes such as vehicle capacities or any other numerical values requiring decimal precision.

Image/Blob: Used to store binary large objects, such as images or documents, associated with entities. For example, a College entity may have a logo image attribute.

4.3.2 Explanation of the data storage and retrieval processes

The data storage and retrieval processes of a college transport management system involve managing and organizing information related to transportation services provided by a college. Here is an explanation of the typical data storage and retrieval processes involved:

Database Design: The system typically utilizes a relational database management system (RDBMS) to store and manage the data. The database design involves identifying the entities, relationships, and attributes relevant to the transportation management system. For example, entities could include students, drivers, vehicles, routes, schedules, and bookings.

Data Collection and Entry: Data related to transportation activities is collected and entered into the system. This can include information such as student details, driver information, vehicle specifications, route information, and scheduling data. The data can be entered manually by administrative staff or through automated processes, such as integrating with student information systems or online registration systems.

Data Storage: The collected data is stored in a structured manner within the database. Each entity is represented as a table, with columns representing attributes and rows containing the actual data entries. For example, there may be separate tables for students, drivers, vehicles, routes, schedules, and bookings, each with their respective attributes and relationships.

Data Retrieval: To retrieve information from the database, queries are used. Queries are requests written in a specific database query language (such as SQL) to retrieve data based on specific criteria or conditions. For example, queries can be used to retrieve information about available routes, schedules for a specific date, or details of students using the transport services.

Data Manipulation: In addition to retrieval, the system may also support data manipulation operations. This includes adding, updating, and deleting data entries in

the database. For instance, administrative staff can add new students, update driver information, assign vehicles to routes, or cancel bookings.

Integration and Reporting: The college transport management system may also integrate with other systems or modules, such as student attendance systems or billing systems, to exchange data and generate reports. This integration allows for a more comprehensive view of transportation-related activities and enables the generation of reports on metrics like student usage, vehicle utilization, or fuel consumption.

Data Security: Data security measures should be implemented to protect the stored information. This can involve using appropriate access controls, encryption methods, and regular data backups to ensure the integrity and confidentiality of the data.

Overall, the data storage and retrieval processes of a college transport management system involve designing a database structure, collecting and entering data, storing it in a structured manner, retrieving information through queries, manipulating data as needed, integrating with other systems, and ensuring data security.

4.3.2 Discussion of the data security and privacy measures implemented in the system

Data security and privacy measures are crucial components of any system, including a college transport management system. Here are some common measures that can be implemented to ensure the security and privacy of the data in such a project:

Access Control: Implementing strong access controls is essential to restrict access to sensitive data. User roles and permissions should be defined, allowing only authorized personnel to access specific data and functionalities. This prevents unauthorized access and minimizes the risk of data breaches.

Authentication and Authorization: Require users to authenticate themselves using strong and unique credentials, such as usernames and passwords, or more advanced authentication methods like two-factor authentication (2FA) or biometrics.

Additionally, implement robust authorization mechanisms to ensure that users only have access to the data they need for their respective roles and responsibilities.

Encryption: Sensitive data, such as personally identifiable information (PII) of students or financial information, should be encrypted both in transit and at rest. Transport Layer Security (TLS) protocols can be used to encrypt data during transmission over networks, and data stored in the database can be encrypted using encryption algorithms to protect it from unauthorized access in case of a breach.

Data Anonymization and De-identification: When possible, sensitive data should be anonymized or de-identified to reduce the risk of identification. This can involve removing or encrypting personally identifiable information (PII) from data sets, using pseudonyms or tokens, and implementing techniques such as data masking or tokenization.

Regular Data Backups: Implementing regular and automated data backups helps ensure that data can be restored in the event of accidental data loss, system failures, or security incidents. Backups should be stored securely and be easily retrievable when needed.

Audit Logs and Monitoring: Enable logging and monitoring mechanisms to track system activities, user access, and data modifications. This allows for the detection of unauthorized access attempts or suspicious activities, aiding in timely response and investigation.

Security Awareness and Training: Conduct regular security awareness programs and training sessions for system administrators, staff members, and users. This helps promote a culture of security awareness and educates individuals about best practices, such as password hygiene, phishing prevention, and handling of sensitive data.

Compliance with Regulations: Ensure that the system adheres to applicable data protection and privacy regulations, such as the General Data Protection Regulation (GDPR) or other local privacy laws. This includes obtaining necessary consents for

data collection and processing, providing mechanisms for data subjects to exercise their rights, and following proper data retention and disposal practices.

Third-Party Vendor Security: If the system integrates with external services or relies on third-party vendors, ensure that proper security measures are in place. Conduct due diligence to assess the security practices of vendors, sign appropriate data protection agreements, and regularly review their compliance with security standards.

Incident Response and Disaster Recovery: Develop and test an incident response plan that outlines procedures for responding to security incidents, such as data breaches or system compromises. Additionally, establish a disaster recovery plan to recover data and resume operations in the event of a catastrophic event or system failure.

Implementing these data security and privacy measures helps mitigate risks, safeguard sensitive information, and maintain the trust of users and stakeholders in the college transport management system. It is important to regularly review and update these measures as new threats and technologies emerge.

CHAPTER -5

Technologies Used

5.1 Front-end

5.1.1 ASP.NET Core MVC

ASP.NET Core MVC is a powerful and versatile framework for building web applications. It follows the Model-View-Controller architectural pattern and provides a structured approach to web project design and development. ASP.NET Core MVC allows you to separate your application's logic into different components such as models, views, and controllers, making your code more maintainable and testable.

Provides robust routing capabilities that allow you to define URL patterns for handling incoming requests. ASP.NET Core MVC also includes features such as model binding, validation, and middleware extensibility. It supports modern development techniques such as dependency injection and asynchronous programming, making it a popular choice for building scalable and performant web applications.

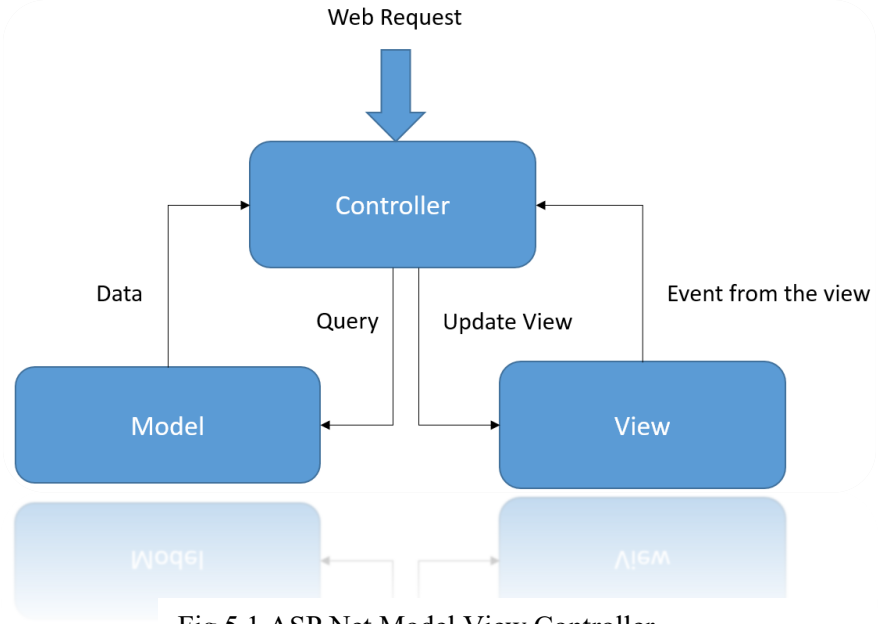


Fig 5.1 ASP.Net Model View Controller

5.1.1 Model

In ASP.NET Core MVC, models represent your application's data and business logic. Defines data structure, behaviors, and validation rules. Models are implemented as classes that encapsulate the application's entities such as users, products, orders, and so on.

They contain properties for displaying data fields and methods for editing and validating data. Models can also define relationships with other models, allowing for complex data structures and associations.

By separating data and business logic into models, developers can achieve a clean and organized architecture. Models work with controllers to process user requests, perform data manipulation, and pass data to views for display. Overall, ASP.NET Core MVC's model plays a key role in data management and implementation of the application's core functionality.

5.1.2 View

In ASP.NET Core MVC, views represent the user interface of your application and are responsible for rendering data received from controllers and making it available to the user. Views are typically implemented as HTML templates with embedded code called Razor syntax that enables dynamic content generation. They define the layout, structure, and visual elements of your application's pages.

Views can access data provided by controllers and display data using various HTML helpers and view components. It also handles user interaction by capturing input and sending it back to the controller for further processing. Views play a key role in providing a visually appealing and interactive experience for users of your web application.

5.1.3 Controller

In ASP.NET Core MVC, controllers act as intermediaries between models and views. It processes user requests, processes data, and determines appropriate responses. Controllers are responsible for receiving incoming HTTP requests, interpreting user intent, and coordinating the necessary actions. It retrieves data from the model, applies business logic, and passes the processed data to the view for rendering.

Controllers also handle form submission, user input validation, and redirects. They play a key role in coordinating the flow of information between the model and the view, allowing developers to implement application behavior and functionality. Controllers represent the central hub for handling user interactions and serve as the backbone of your application logic.

5.2 HTML

Hypertext Markup Language (HTML) is the usual markup language for files designed to be displayed in an internet browser. It can be assisted by technology such as Cascading Style Sheets (CSS) and scripting languages along with JavaScript. Web browsers obtain HTML files from an internet server or from a local garage and render the files into multimedia internet pages. HTML describes the shape of a web optioned page semantically and at first blanketed cues for the advent of the record.

HTML elements are the constructing blocks of HTML pages. With HTML constructs, pictures and other items consisting of interactive paperwork may be embedded into the rendered web page. HTML offers a means to create based documents by denoting structural semantics for textual content including headings, paragraphs, lists, hyperlinks, costs and different gadgets.

HTML factors are delineated by way of tags, written the use of perspective brackets. Tags including and at once introduce content into the page. Other tags which include surround and offer data approximately document text and may consist of other tags as sub-factors.

Browsers do not display the HTML tags, however use them to interpret the content material of the web page. HTML can embed packages written in a scripting language such as JavaScript, which influences the behaviour and content of internet pages. Inclusion of CSS defines the look and layout of content.

The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has advocated using CSS over specific presentational HTML due to the fact 1997.

5.3 Cascading Style Sheet

Cascading Style Sheets (CSS) is a fashion sheet language used for describing the presentation of a record written in a markup language like HTML. CSS is a cornerstone generation of the World Wide Web, alongside HTML and JavaScript. 19 CSS is designed to permit the separation of presentation and content, inclusive of layout, colourings, and fonts. This separation can improve content accessibility, offer greater flexibility and control in the specification of presentation characteristics, permit a couple of web pages to percentage formatting through specifying the relevant CSS in a separate .css record, and reduce complexity and repetition within the structural content.

CSS information can be furnished from various resources. These assets may be the net browser, the user and the writer. The statistics from the author can be further labelled into inline, media kind, importance, selector specificity, rule order, inheritance and property definition.

CSS style data may be in a separate document or it may be embedded into an HTML record. Multiple Fashion sheets may be imported. Different patterns may be implemented depending on the output device being used; for example, the display model may be pretty distinctive from the published version, so that authors can tailor the presentation as it should be for every medium.

The fashion sheet with the highest precedence controls the content material shown. Declarations not set within the maximum priority source are exceeded directly to a

source of decreased priority, such as the person agent fashion. The process is called cascading.

5. 4 JavaScript

JavaScript is a high-stage, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-primarily based item-orientation, and satisfactory functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive internet pages and is a crucial part of net programs.

The enormous majority of websites use it, and most important net browsers have a dedicated JavaScript engine to execute it, and a multi-paradigm language, JavaScript supports event-driven, purposeful, and imperative (along with item-oriented and prototype-based totally) programming patterns.

It has APIs for operating with text, arrays, dates, everyday expressions, and the DOM, but the language itself does not consist of any I/O, which includes networking, garage, or portraits facilities.

It relies upon the host environment in which it's far embedded to offer those features. 20 Initially handiest carried out client-facet in net browsers, JavaScript engines are actually embedded in lots of other kinds of host software, together with server side in internet servers and databases, and in non-net packages including phrase processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and computer programs, along with desktop widgets.

Backend

5.6 C#

C# (pronounced cis) is a modern general-purpose programming language developed by Microsoft. Designed to be simple, efficient and versatile, it is suitable for a wide variety of application development scenarios. C# is part of the .NET Framework and is widely used for developing Windows applications, web services, and enterprise software. It combines the power of object-oriented programming with features such as type safety,

garbage collection, and automatic memory management.

C# supports different programming paradigms, including imperative, declarative, and functional programming. It offers a rich set of libraries and frameworks that make it easy to build robust and scalable applications. C# remains a popular choice for developers around the world due to its strong community support and continuous evolution.

5.7 MySQL

MySQL is an open-source relational database management device (RDBMS) primarily based on Structured Query Language (SQL). It Is one a part of the very famous LAMP platform along with Linux, Apache, My SQL, and PHP. Currently My SQL is owned with the aid of Oracle. My SQL database is available on most essential OS structures. It Runs on BSD Unix, Linux, Windows, or Mac OS. Wikipedia and YouTube use My SQL. These websites control millions of queries each day. My SQL is available in versions: My SQL server system and My SQL embedded system.

5.8 RDBMS Terminology

Before we continue to provide an explanation for the MySQL database device, let's revise a few definitions associated with databases.

● Database: A database is a collection of tables, with associated information.

● Table: A table is a matrix with facts. A desk in a database seems like an easy spreadsheet.

● Column: One column (statistics element) incorporates information of 1 and the equal kind, for example the column postcode.

● Row: A row (tuple, entry or file) is a group of related facts, for instance the data of 1 subscription.

● Redundancy: Storing information twice, redundantly to make the system quicker.

● Primary Key: A primary key is specific. A key value can't occur two times in one desk. With a key, you could find at most one row.

● Foreign Key: A foreign secret's the linking pin between two tables.

● Compound Key: A compound key (composite key) is a key that consists of multiple columns, due to the fact one column is not sufficiently particular.

● Index: An index in a database resembles an index behind a book.

● Referential Integrity: Referential Integrity makes sure that an overseas key cost continually factors to a current row.

Chapter 6

Module Description

This is a Transport Management System website that allows the admin to manage buses and student details. The admin can add and delete buses, as well as add and delete student details. The admin can also allot the buses to students and track the location of the buses in real-time. The website also features a student portal, where students can view their generated bus pass and track the location of the buses. The website is designed to be user-friendly and customizable to meet the specific needs of each user. The goal of the website is to streamline transportation operations, reduce costs, and improve efficiency.

The project entitled “Transport Management System” mainly broken into following modules.

- Admin module
- Student module
- Driver module
- Notice module
- Validation and Security

These modules are meant to perform different functionality for the Admin end and User end.

Admin Module

- Admin is the one having overall control of the website.
- Admin LOGIN using username and password.

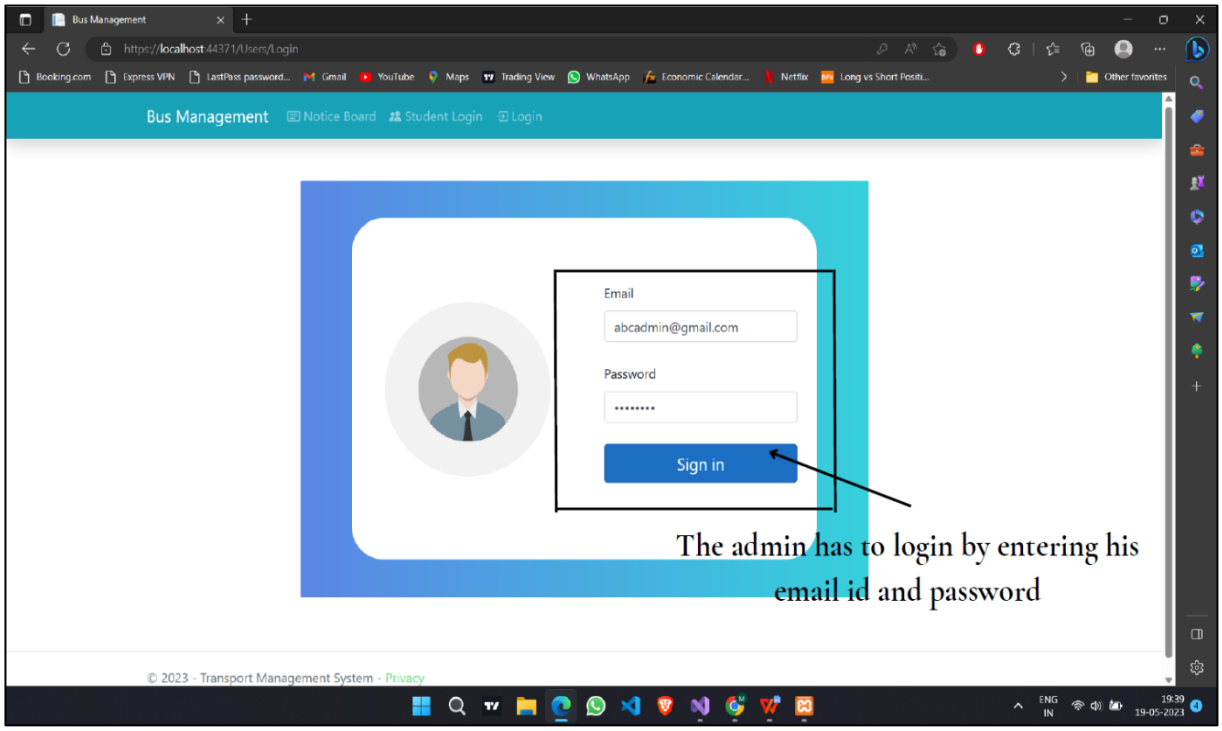


Fig 5.2 Login Page

- Admin is one who can add, delete or modify the buses.

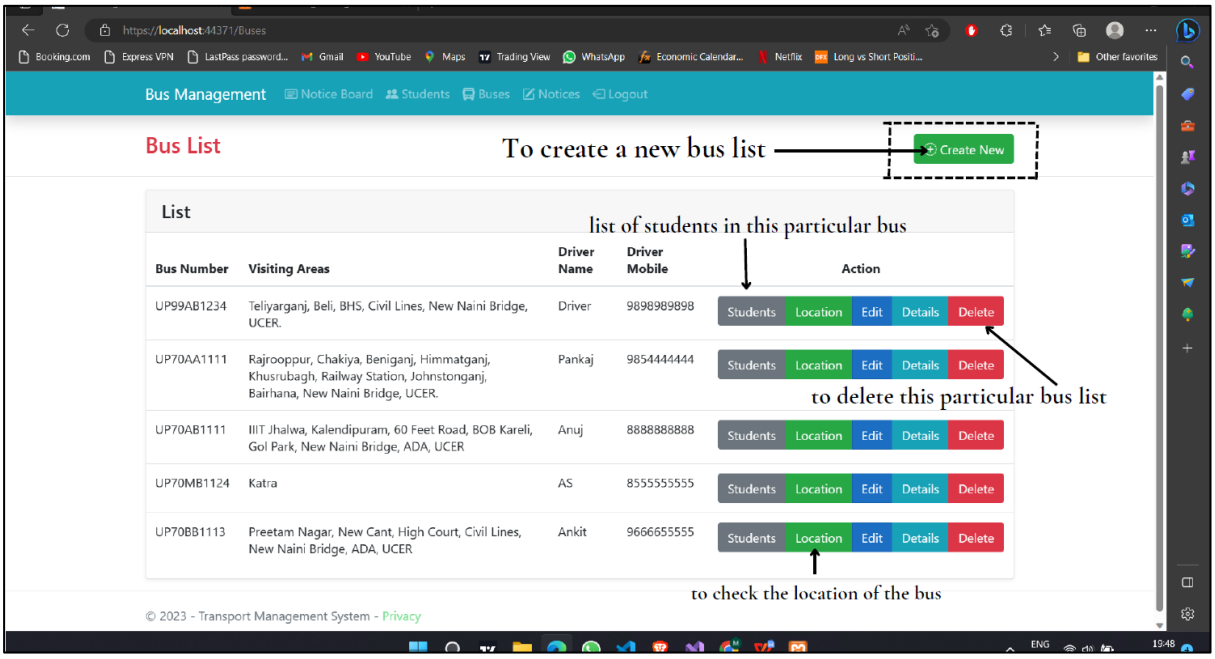


Fig 5.3 Bus Management List

- When the student pays the bus fees, admin adds the student to the database.

- Admin is responsible for adding, deleting, or modifying the students.

- Admin allots the buses to the students as per their addresses and bus routes.

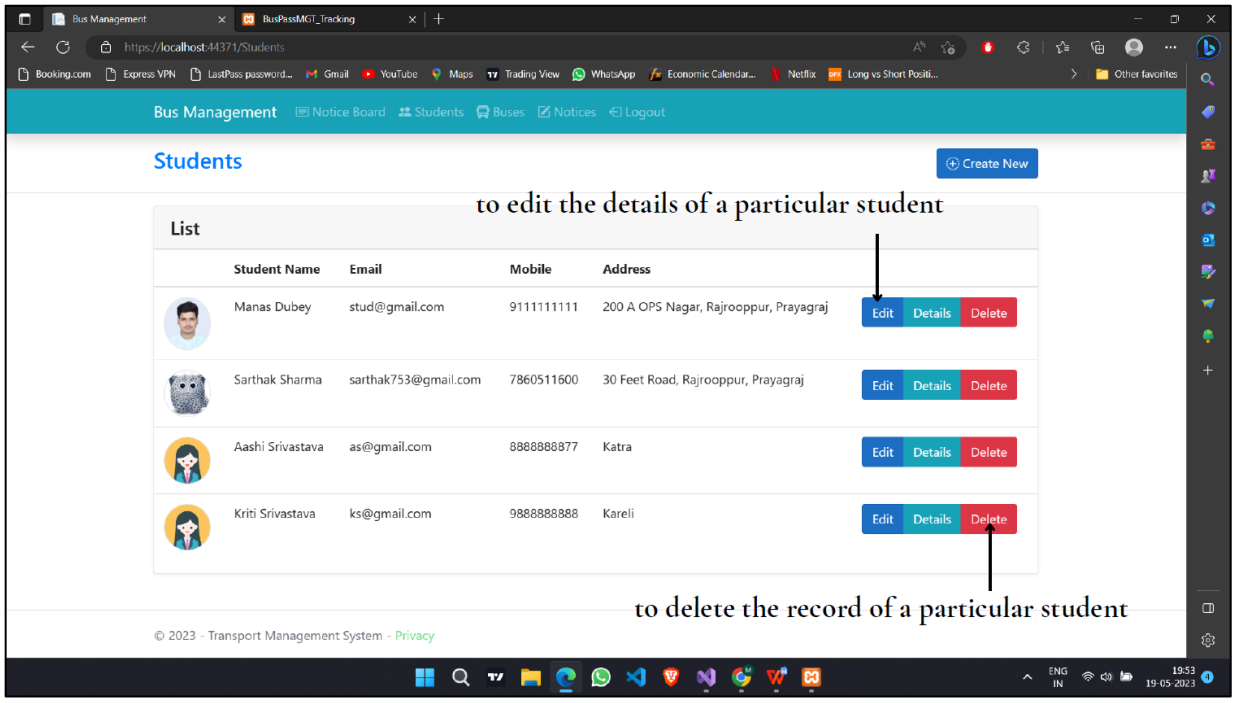


Fig 5.4 List of Students

- Admin can add, delete, or edit notices and publish them to the notice board.

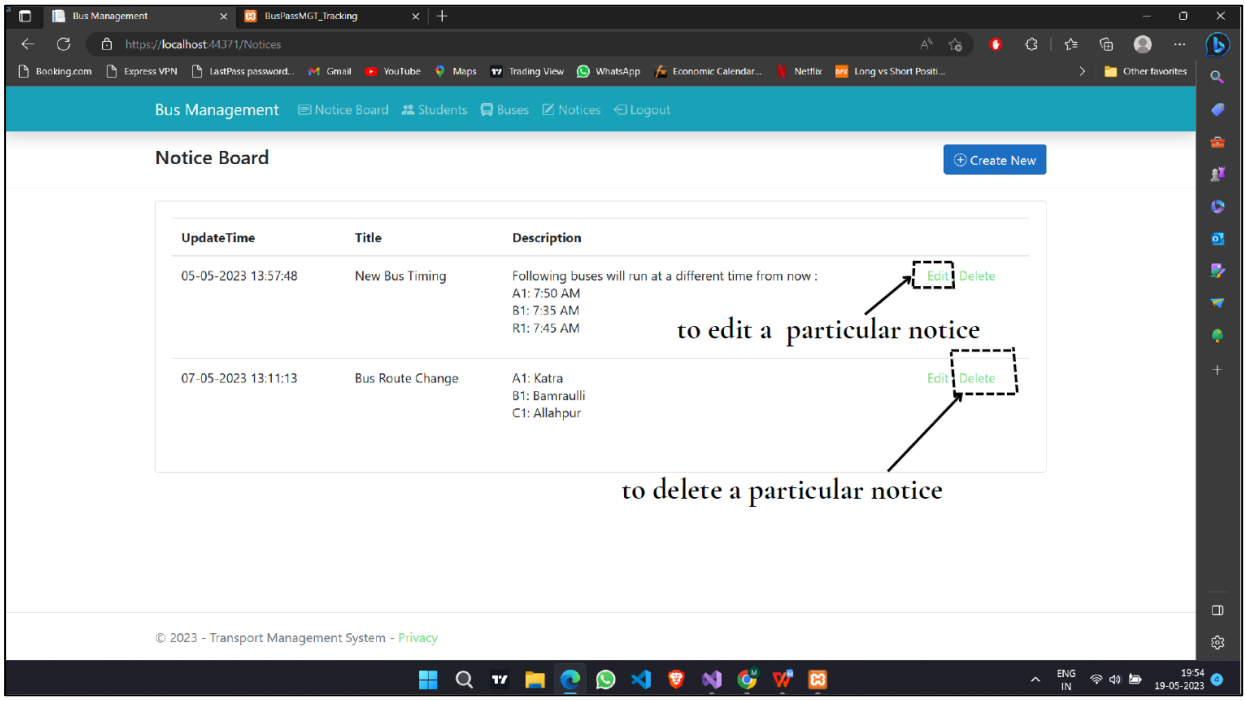


Fig 5.5 Notice Board

Student Module

- Students LOGIN using username and password.

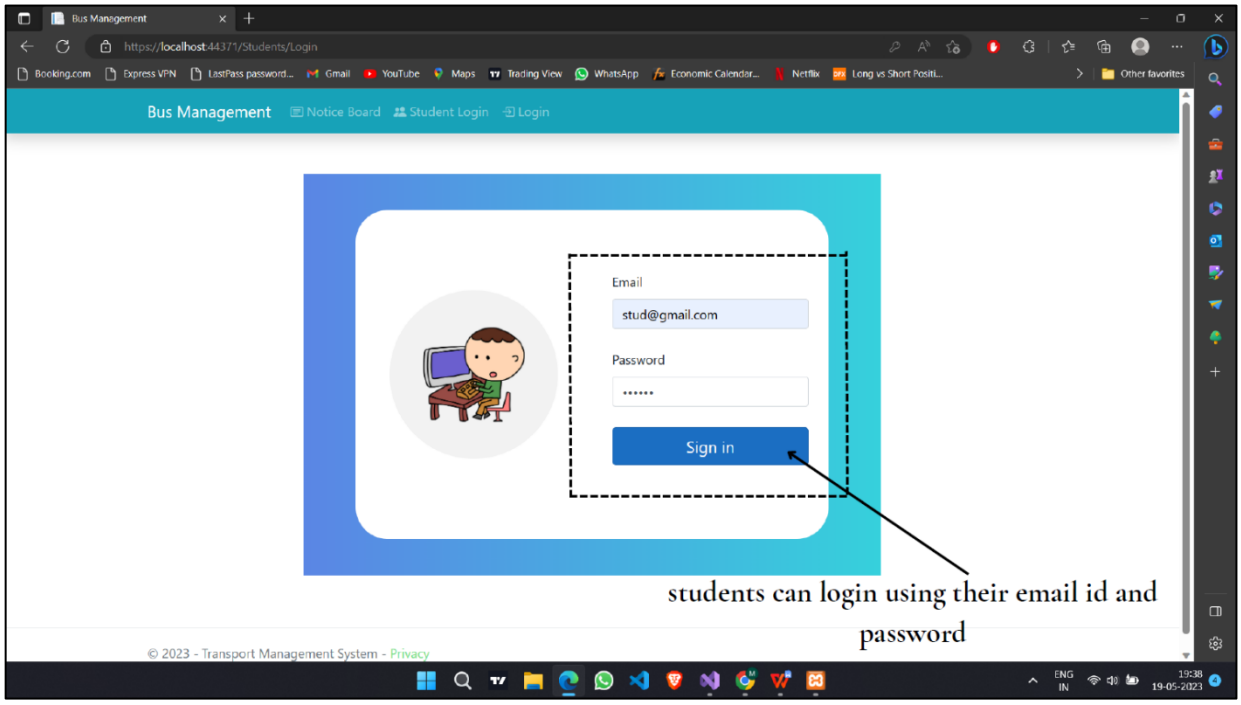


Fig 5.6 Student Login Page

- For the first time, the student's credentials are his or her email address, and the password is his or her mobile phone number.

- Students can change the password and need to verify the old password then new password with confirm password.

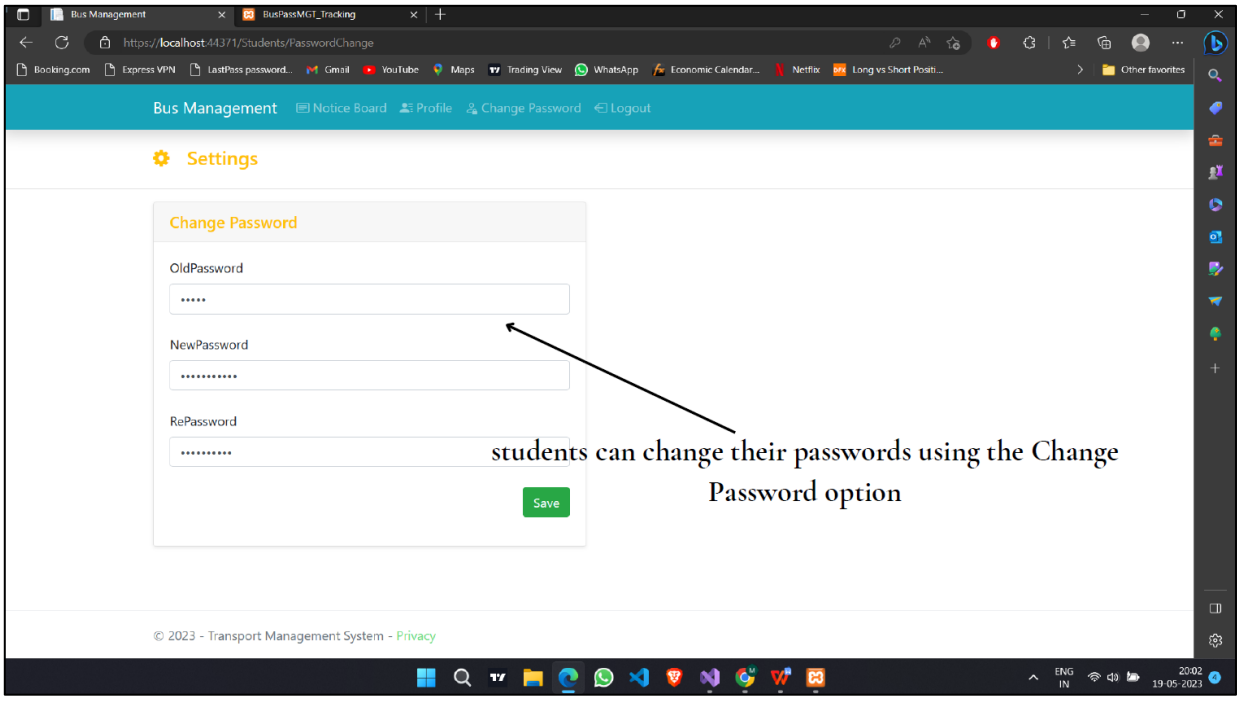
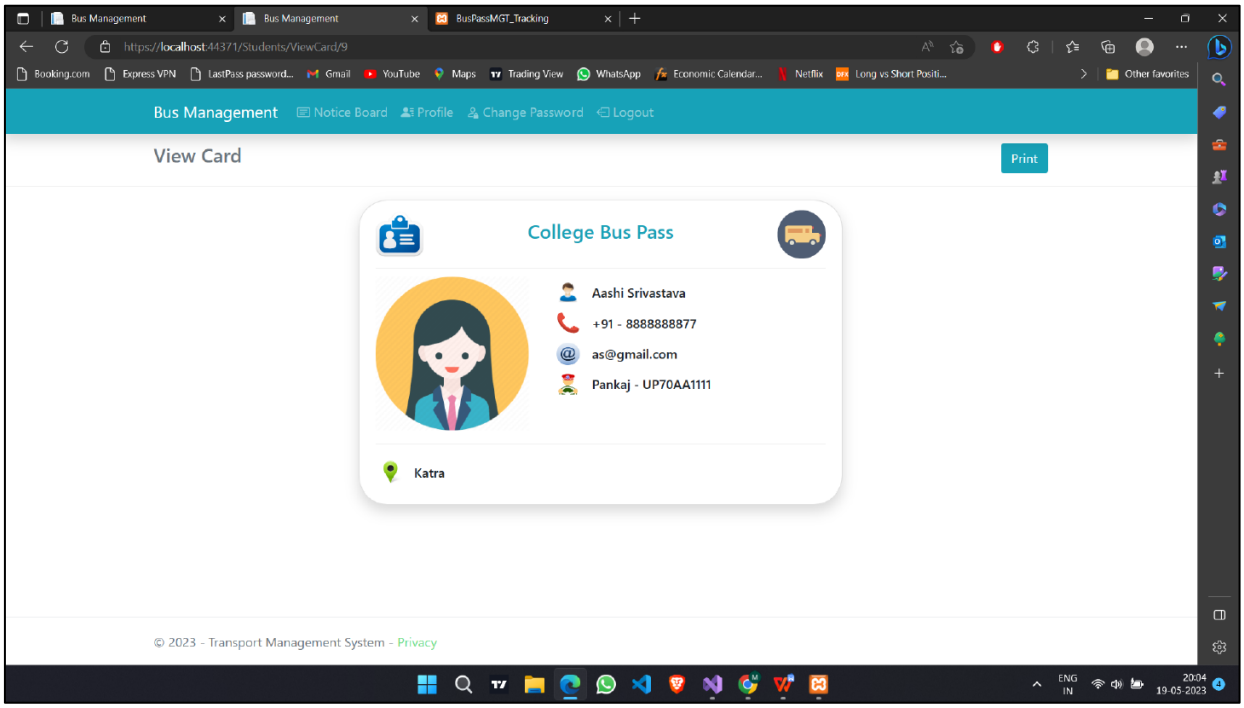


Fig 5.7 Change Password

- Students can view their details.
- Students can view the generated bus pass and use the same for verification while travelling on buses.



5.8 Bus Pass

- Students can check the current location of their allotted buses.

- Students can check the notices published by using the notice board tab.

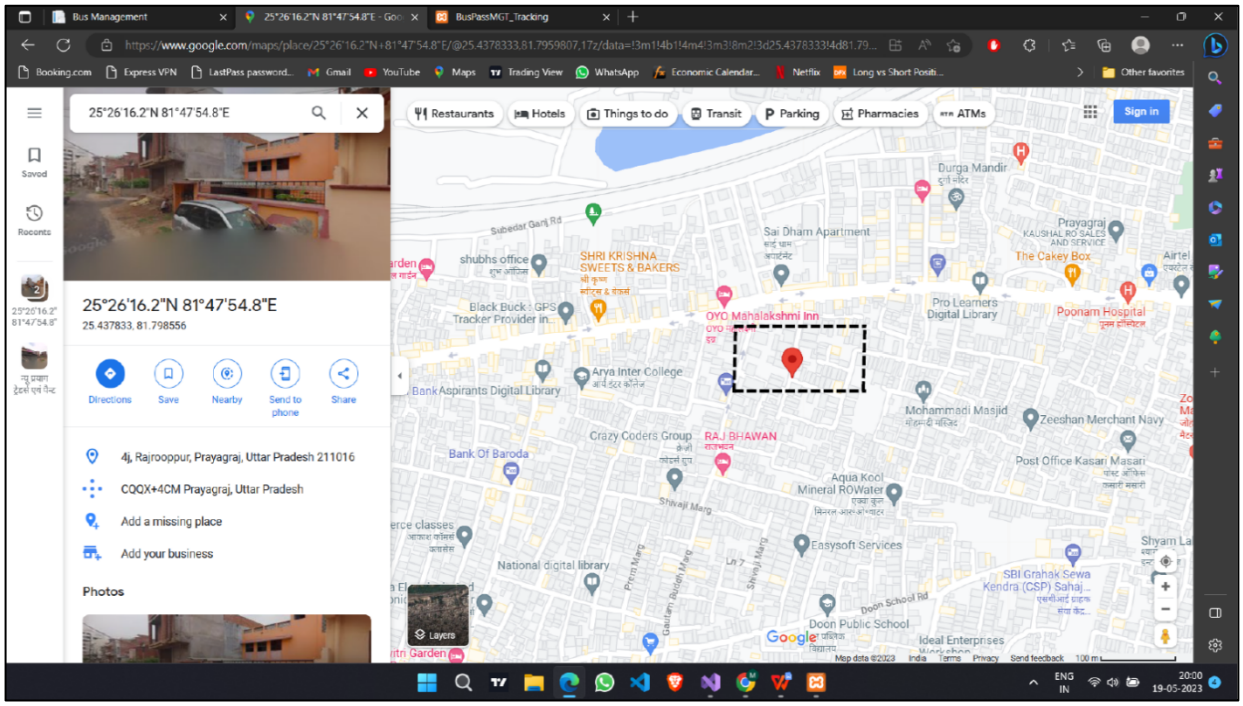


Fig 5.9 Map Location

Driver Module

- A driver can login using the bus registration number and his or her mobile number from GPS-enabled devices like mobile phones.

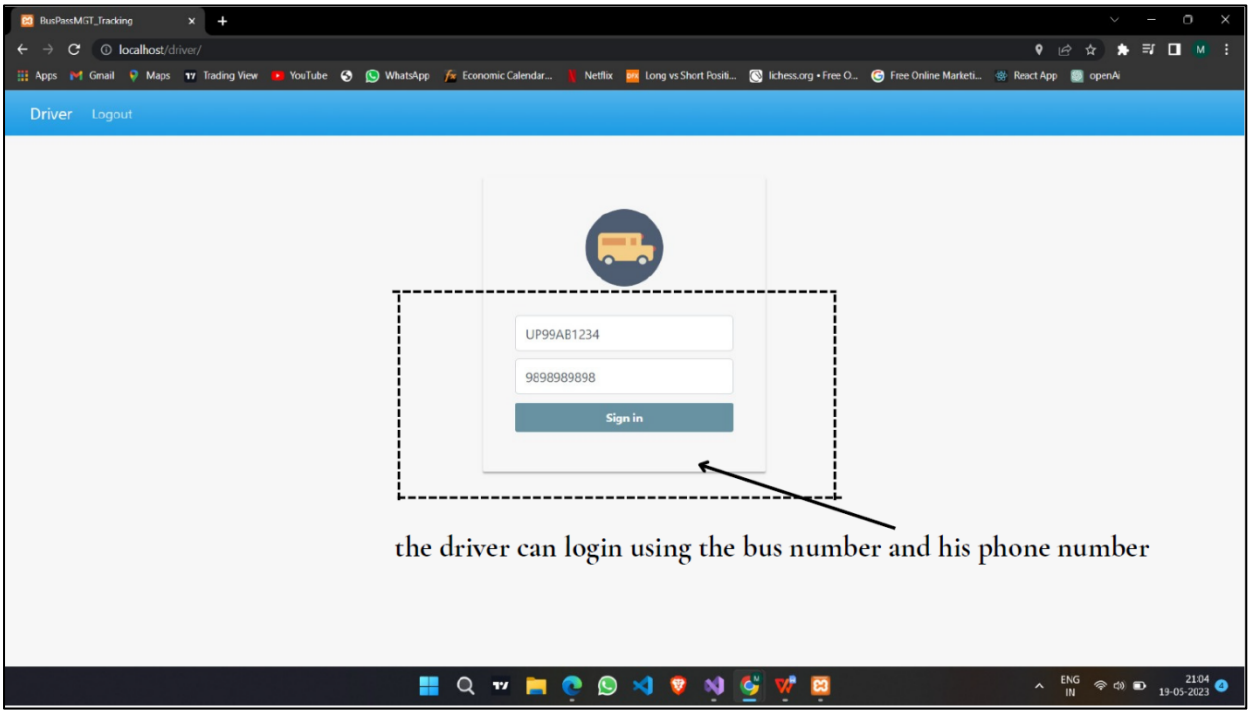


Fig 5.10 Driver Login

- He or she can START/STOP tracking the buses by clicking on the button provided.

- Once pressed, the device starts sending coordinates, which can be used to fetch the location.

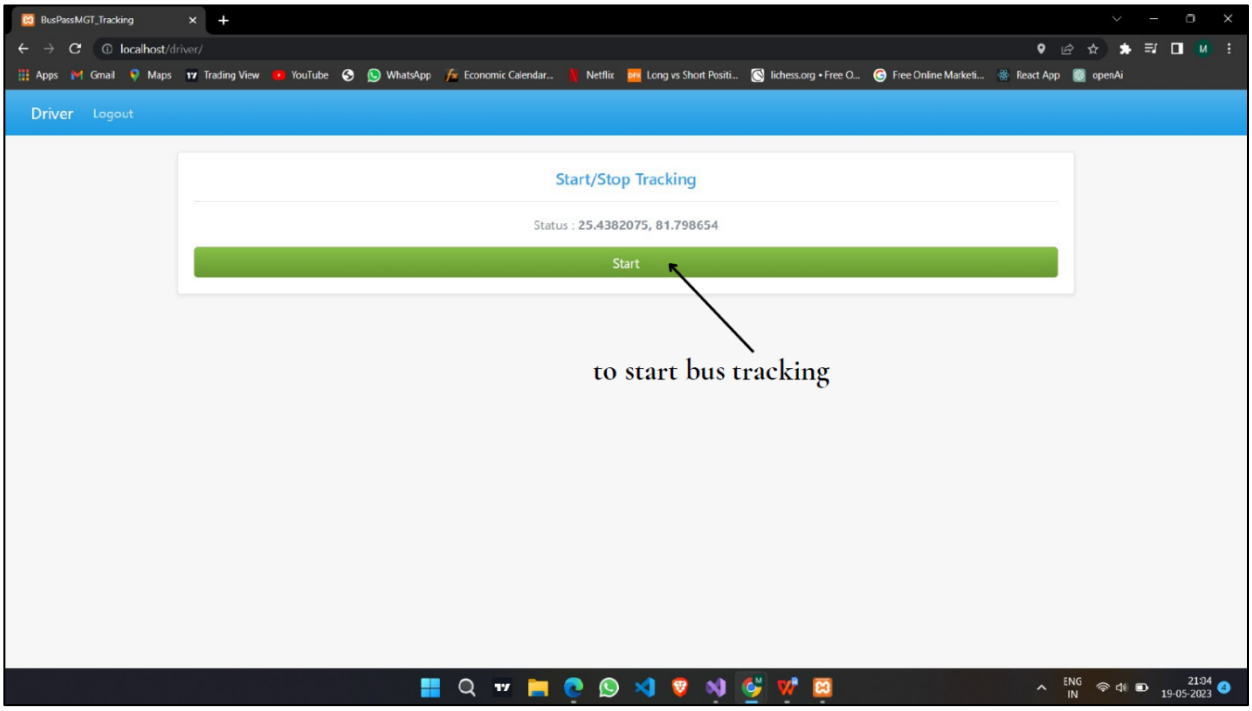


Fig 5.11 Start Tracking

Notice Module

- The Notice Board module is designed to provide a centralized location for all notices that need to be shared with the community.

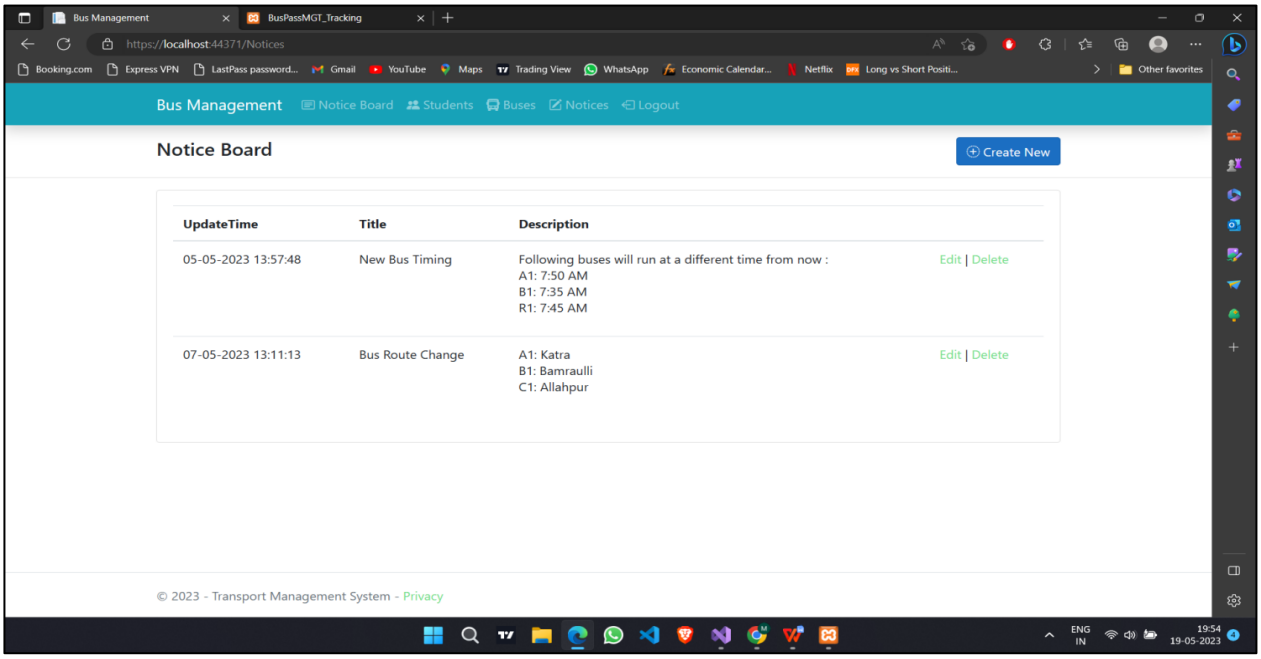


Fig 5.12 Notice Module

- Admins have full control over the content that is posted on the Notice Board and can easily add, modify, or delete notices as needed.

- The Notice Board is accessible to everyone in the community, making it easy for individuals to stay up-to-date on important news and events.

Validation and Security

- SSL encryption is a key component of our transport management system. It ensures that sensitive user data, such as user credentials and location information, is protected from unauthorized access.
- Validation techniques are used to ensure that user data is accurate and complete. Server-side validation of user inputs and data checks are used to ensure that location data is within expected ranges. This helps ensure that the system is reliable and secure for all users.
- Our transport management system includes access controls and user authentication mechanisms ensure that only authorized personnel can access user data and modify system settings.

Implementation and Testing

I. Implementation

A. Develop Implementation Plan

1. Define scope of implementation: This involves identifying what needs to be implemented, which in this case is the four main modules of the project, and specifying the technologies that will be used to implement them.

Modules:

- Admin Module
- Student Module
- Driver Module
- Notice Module

Technology:

- ASP.NET
- CSS
- Bootstrap
- AngularJS
- MySQL

2. Develop detailed schedule: This involves creating a timeline for each module and assigning tasks to team members to ensure that the project is completed on time.
3. Identify resources and assign tasks: This involves assigning developers to each module and providing them with the necessary resources such as software and hardware.

B. Prepare for Implementation

Install necessary software and hardware: This involves installing the software and hardware that will be used to implement each module.

Software :

- XAMPP
- Visual Studio
- Google Chrome
- Microsoft edge

Hardware :

- CPU: Intel Core i3, i5, i7,i9, AMD Ryzen 3, 5, 7 with 2 GHz minimum OR Android 6.0 or above with GPS technology
- RAM: 2GB or above
- Hard Disk Space: 2GB minimum free.
- Input/Output Devices: Keyboard, Mouse, Monitor

2. Configure system settings: This involves configuring system settings for each module to ensure that it works correctly.
3. Train personnel: This involves training developers on how to implement each module.

C. Execute Implementation Plan

1. Develop and execute test cases: This involves developing test cases for each module and executing them to ensure that each module is working correctly.
2. Verify system functionality: This involves verifying that each module is working correctly and that all features are available.
3. Deploy system: This involves deploying each module to the server.

D. Monitor Implementation

1. Monitor system performance: This involves monitoring system performance to ensure that each module is working correctly.
2. Resolve issues and errors: This involves resolving any issues and errors that arise during implementation.
3. Provide support to personnel: This involves providing support to developers during implementation.

II. Testing

A. Develop Test Plan

1. Define scope of testing: This involves identifying what needs to be tested, which in this case is each module, and specifying the types of tests that will be conducted.
2. Develop a detailed schedule: This involves creating a timeline for each module and assigning tasks to testers to ensure that the project is tested thoroughly.
3. Identify resources and assign tasks: This involves assigning testers to each module and providing them with the necessary resources such as test cases.

B. Prepare for Testing

1. Develop test cases: This involves developing test cases for each module to ensure that it works correctly.
2. Configure test environment: This involves configuring the test environment for each module to ensure that it is set up correctly.

C. Performance and Scalability Testing

Measure response times, latency, and resource consumption to identify potential bottlenecks and optimize system performance.
Test the system's performance under different loads and conditions to ensure it can handle concurrent attendance requests and database interactions.
Evaluate the scalability of the system by simulating increased user volumes and attendance requests to ensure it can scale without compromising performance and functionality.

D. User Acceptance Testing

Approach: The approach for the UAT will be to conduct a series of tests to ensure that each module is working correctly and that all features are available.

Test Schedule: The UAT will be executed by the testing team. Testers will be assigned to each module and will execute the test cases.
Test Results: Test results will be recorded and tracked in a test management tool. Any issues or defects will be reported to the development team for resolution.
Sign-off: The project sponsor or project manager will sign off the UAT. The sign-off will indicate that the UAT has been completed and that each module meets the acceptance criteria.

E. Deployment and Implementation

Create a staging environment to test the software before deployment. Use version control to manage the codebase and ensure that the latest version is deployed.
Test the software in the staging environment to ensure that it works correctly. Once the software is tested and approved, deploy it to the production environment.
Monitor the production environment for issues and take corrective action as needed. Create a rollback plan in case of issues with the deployment.
Communicate the deployment schedule and any potential downtime to stakeholders. Provide training to users on how to use the new software.
Ensure that the software is compliant with all necessary regulations and licensing agreements. Create a plan for ongoing maintenance and support of the software.

Conclusion

- Implementing a bus pass generation system simplifies the process of issuing and managing student bus passes. This eliminates manual paperwork, reduces administrative burden, and provides a convenient digital solution.

- With increasing focus on student safety, efficient transportation, and digitization of processes in educational institutions, there is a growing demand for reliable and advanced bus management systems. The project can cater to this demand and potentially attract interest from schools, colleges, or other organizations.

- With the TMS, you can get real-time information about transportation options, arrival times, and any changes. This helps you plan your trips better and reduces waiting time.

- The TMS is designed to be user-friendly. It has a web app tracking vehicles, and getting updates is easy. It also helps communication between drivers, students, and staff.

- Overall, the TMS is a smart system that improves transportation on campus. It saves time, keeps you informed, prioritizes safety, and considers the environment.

Future Scope

- By creating dedicated mobile applications for students, parents, and bus drivers, we can improve the user experience and offer additional features such as push notifications and route suggestions.

- Continually improving security measures include features such as bus boarding, geofencing to monitor bus movement, and integration with access control systems.

- The development of comprehensive analytics and reporting modules provides valuable insights into bus utilization, performance and cost optimization to aid decision-making and future planning.

- TMS can use advanced technologies such as artificial intelligence and smart devices to make it even smarter and more efficient. It could learn from data and make better decisions about routes and schedules.

- TMS can be further customized to suit your needs. May remember user preferences and offer suggestions based on previous selections. This makes using the system more convenient and individual.

- TMS can use data analytics to make smarter decisions. You can study patterns and trends to predict demand and optimize your services. This helps improve the overall efficiency and effectiveness of transportation.

- Integration with smart cities: As cities become smarter, TMS may be integrated with other systems within the city, such as traffic management and urban planning efforts. This creates a more connected transportation system and enhances the overall city experience. By leveraging these opportunities, the TMS of the future aims to deliver a smarter, more convenient and more sustainable transportation experience.

References

• ASP.NET Core MVC Framework - Dot Net Tutorials

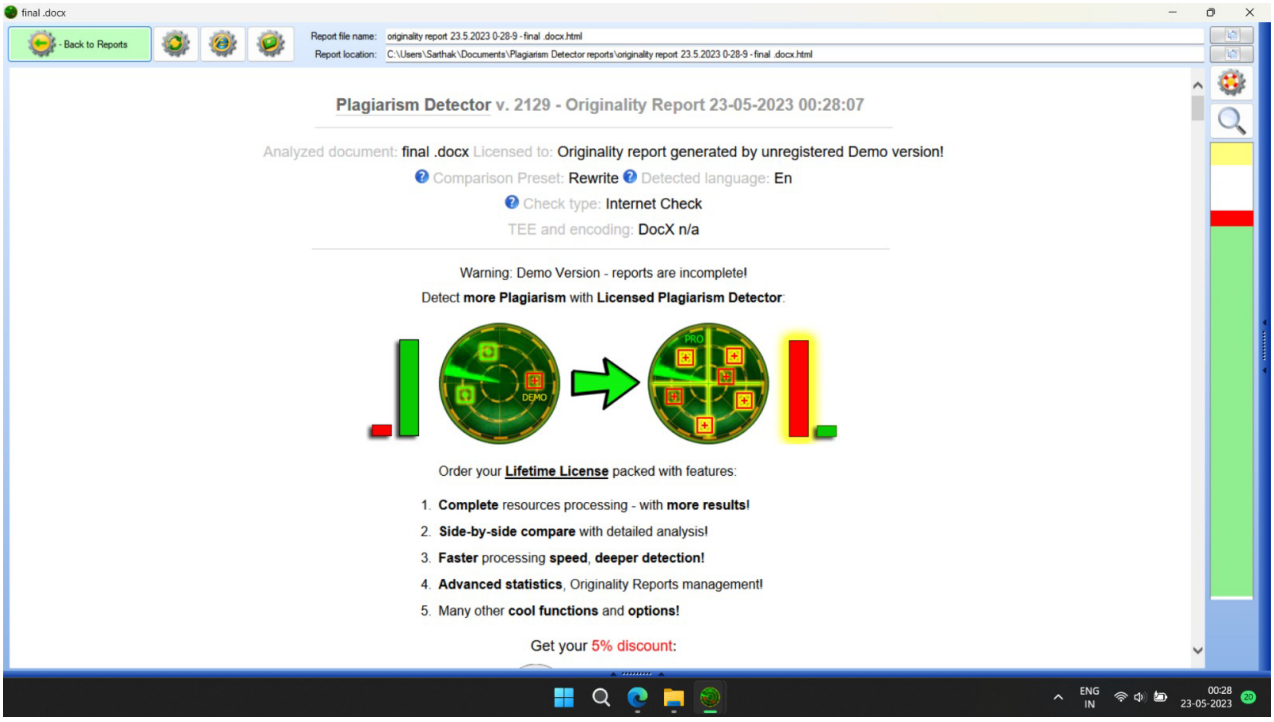
• Icons in ASP.NET MVC Component (syncfusion.com)

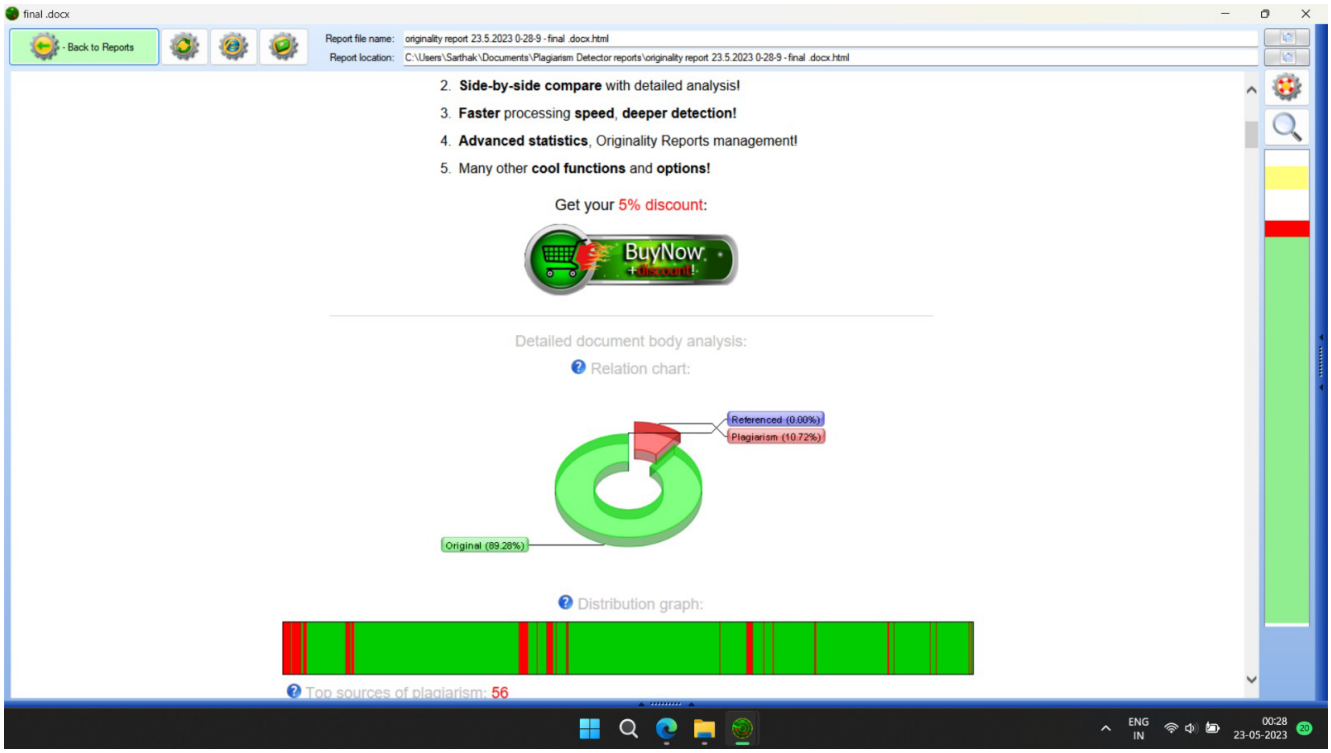
• Asp.Net Core MVC 3.1 course overview | Asp.Net Core tutorial - YouTube

• Learn ASP.NET Core 6.0 - Full Course for Beginners[Tutorial] in Hindi | Day-1 - YouTube

• https://dev.to/charkinsdevelopment/how-to-build-a-toastr-js-notification-system-using-asp-net-mvc-1f97

PLAGIARISM REPORT




final .docx
Back to Reports
Report file name: originality report 23.5.2023 0-28-9 - final .docx.html
Report location: C:\Users\Sarthak\Documents\Plagiarism Detector reports\originality report 23.5.2023 0-28-9 - final .docx.html
2. Side-by-side compare with detailed analysis!
3. Faster processing speed, deeper detection!
4. Advanced statistics, Originality Reports management!
5. Many other cool functions and options!
Get your 5% discount:
BuyNow
Detailed document body analysis:
Relation chart:
Referenced (0.00%)
Plagiarism (10.72%)
Original (89.28%)
Distribution graph:
Top sources of plagiarism: 56
ENG
IN
00:28
23-05-2023