**AJCETRIPS**

*Project Report Submitted by*

**AJUL K JOSE**

**Reg. No.: AJC00MCA-I004**

*In Partial fulfillment for the Award of the Degree Of*

**INTEGRATED MASTER OF COMPUTER APPLICATIONS**

**(INMCA)**

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING**

**KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with ‘A’ grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

# 2024-2025

## DEPARTMENT OF COMPUTER APPLICATIONS

### AMAL JYOTHI COLLEGE OF ENGINEERING

**KANJIRAPPALLY**



**CERTIFICATE**

This is to certify that the Mini Project report, “NEWZILLA**”** is the bona fide work of **ROHAN KURIAN MATHEW (Regno: AJC00MCA-I049)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2022-23.

**Mr. Binumon Joseph Mr. Binumon Joseph**

**Internal Guide Coordinator**

**Rev. Fr. Dr. Rubin Thottupurathu Jose**

**Head of the Department**

**DECLARATION**

I hereby declare that the project report **“AJCETRIPS”** is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2024-2025.

**Date: 26/10/2024 AJUL K JOSE**

**KANJIRAPPALLY Reg: AJC00MCA-I004**

# ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Manager **Rev. Fr. Dr. Roy Pazhayaparampil** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Mr. Binumon Joseph** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Mr. Binumon Joseph** for his inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

AJUL K JOSE

# ABSTRACT

.

AJCETRIPS is an advanced platform designed to revolutionize college campus operations through automation and innovative technology integration. The platform includes a Flutter Android application for College Vehicle System, providing users with seamless access to essential campus services. The College Vehicle Management module is a Flutter Android app exclusively accessible to college staff. It allows drivers to efficiently manage vehicle details, such as insurance information, test dates, maintenance schedules, and trip details. Administrators can monitor this data in real-time through a web interface, ensuring the efficient upkeep and operation of college-owned vehicles and tracking trip information for better resource management.

**CONTENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL. NO** | | **TOPIC** | **PAGE NO** | |
| **1** | | **INTRODUCTION** | **1** | |
| **1.1** | | **PROJECT OVERVIEW** | **2** | |
| **1.2** | | **PROJECT SPECIFICATION** | **2** | |
| **2** | | **SYSTEM STUDY** | **4** | |
| **2.1** | | **INTRODUCTION** | **5** | |
| **2.2** | | **EXISTING SYSTEM** | **5** | |
| **2.3** | | **DRAWBACKS OF EXISTING SYSTEM** | **7** | |
| **2.4** | | **PROPOSED SYSTEM** | **7** | |
| **2.5** | | **ADVANTAGES OF PROPOSED SYSTEM** | **7** | |
| **3** | | **REQUIREMENT ANALYSIS** | **9** | |
| **3.1** | | **FEASIBILITY STUDY** | **10** | |
| **3.1.1** | | **ECONOMICAL FEASIBILITY** | **10** | |
| **3.1.2** | | **TECHNICAL FEASIBILITY** | **10** | |
| **3.1.3** | | **BEHAVIORAL FEASIBILITY** | **11** | |
| **3.1.4** | | **FEASIBILITY STUDY QUESTIONNAIRE** | **11** | |
| **3.2** | | **SYSTEM SPECIFICATION** | **12** | |
| **3.2.1** | | **HARDWARE SPECIFICATION** | **12** | |
| **3.2.2** | | **SOFTWARE SPECIFICATION** | **12** | |
| **3.3** | | **SOFTWARE DESCRIPTION** | **13** | |
| **3.3.1** | | **PHP** | **13** | |
| **3.3.2** | | **MYSQL** | **13** | |
| **4** | | **SYSTEM DESIGN** | **14** | |
| **4.1** | | **INTRODUCTION** | **15** | |
| **4.2** | | **UML DIAGRAM** | **15** | |
| **4.2.1** | | **USE CASE DIAGRAM** | **16** | |
| **4.2.2** | | **SEQUENCE DIAGRAM** | **17** | |
| **4.2.3** | | **STATE CHART DIAGRAM** | **18** | |
| **4.2.4** | | **ACTIVITY DIAGRAM** | **18** | |
| **4.2.5** | | **CLASS DIAGRAM** | **19** | |
| **4.2.6** | | **OBJECT DIAGRAM** | **19** | |
| **4.2.7** | | **COMPONENT DIAGRAM** | **20** | |
| **4.2.8** | | **DEPLOYMENT DIAGRAM** | **21** | |
| **4.2.9** | | **COLLABORATION DIAGRAM** | **21** | |
| **4.3** | | **USER INTERFACE DESIGN USING FIGMA** | **22** | |
| **4.4** | | **DATA BASE DESIGN** | **24** | |
| **5** | | **SYSTEM TESTING** | **29** | |
| **5.1** | | **INTRODUCTION** | **30** | |
| **5.2** | | **TEST PLAN** | **30** | |
| **5.2.1** | **UNIT TESTING** | | **31** |
| **5.2.2** | **INTEGRATION TESTING** | | **32** |
| **5.2.3** | **VALIDATION TESTING** | | **33** |
| **5.2.4** | **USER ACCEPTANCE TESTING** | | **34** |
| **5.2.5** | **AUTOMATION TESTING** | | **34** |
| **5.2.6** | **SELENIUM TESTING** | | **36** |
| **6** | **IMPLEMENTATION** | | **49** |
| **6.1** | **INTRODUCTION** | | **50** |
| **6.2** | **IMPLEMENTATION PROCEDURE** | | **50** |
| **6.2.1** | **USER TRAINING** | | **52** |
| **6.2.2** | **TRAINING ON APPLICATION SOFTWARE** | | **53** |
| **6.2.3** | **SYSTEM MAINTENANCE** | | **54** |
| **7** | **CONCLUSION & FUTURE SCOPE** | | **56** |
| **7.1** | **CONCLUSION** | | **57** |
| **7.2** | **FUTURE SCOPE** | | **57** |
| **8** | **BIBLIOGRAPHY** | | **59** |
| **9** | **APPENDIX** | | **61** |
| **9.1** | **SAMPLE CODE** | | **62** |
| **9.2** | **SCREEN SHOTS** | | **76** |

## List of Abbreviation

HTML - Hyper Text Markup Language

CSS - Cascading Style Sheet

PHP - Hypertext Preprocessor

SQL - Structured Query Language

jQuery - Javascript Query

AJAX - Asychronous JavaScript and XML

UML - Unified Modeling Language

# CHAPTER 1

# INTRODUCTION

### PROJECT OVERVIEW

This project, AJCETRIPS, is designed to revolutionize college campus vehicle operations by leveraging automation and integrating innovative technology. The platform includes a Flutter Android application for the College Vehicle Management System that provides seamless access to essential services related to campus transportation. This system is exclusively accessible to college staff, including drivers and administrators, allowing efficient management and monitoring of vehicle data.

The College Vehicle Management Module helps users track vehicle information such as insurance, test dates, maintenance schedules, and trip details. The platform ensures that college-owned vehicles are well-maintained, and administrators can monitor these operations through a real-time web interface for improved resource management.

**Users:**

* Admin:
  + Manage vehicle details such as insurance, maintenance, and test dates.
  + Monitor and update trip details.
  + Oversee real-time vehicle status and resource allocation.
  + Receive notifications for important updates (e.g., insurance expiry, upcoming maintenance).
* Driver:
* View vehicle details, including insurance, test dates, and maintenance schedules.
* Add trip details for college vehicles.
* Receive notifications for important updates (e.g., insurance expiry, upcoming maintenance).
* Normal User:
* View general vehicle details and trip information.
* Access to limited information relevant to general users of the system, but without the ability to make changes.

### PROJECT SPECIFICATION

AJCETRIPS provides a unified platform for managing college vehicle operations efficiently through a Flutter Android App. The system streamlines vehicle management, maintenance tracking, and trip monitoring for staff, ensuring timely and organized vehicle use.

The College Vehicle Management Module within the app is user-friendly and designed for drivers and administrators to manage vehicle details, receive notifications, and oversee campus transportation operations.

Key system features include:

❖ Admin Dashboard: The app includes an Admin Dashboard, granting administrators full control over vehicle data, such as adding or updating insurance details, fitness test dates, and maintenance records. Admins can also monitor trip logs, generate vehicle usage reports, and manage notifications related to upcoming maintenance or document expirations.

❖ Driver Dashboard: The app provides drivers with access to view vehicle details, including insurance status and maintenance schedules. Drivers can also add trip details, log usage, and receive real-time notifications about insurance expirations, fitness test dates, and scheduled maintenance.

❖ Real-Time Notifications: Through Firebase Cloud Messaging, drivers and admins receive automatic notifications for important updates (insurance expirations, fitness test due dates, or upcoming maintenance), ensuring that vehicles remain operational and compliant with regulations.

❖ Resource Management and Analytics: The Admin Dashboard offers resource management tools to track vehicle performance, monitor trip histories, and manage the overall usage of college vehicles. This helps in ensuring optimal utilization of resources and minimizing vehicle downtime.

❖ Feedback Mechanism: The platform allows drivers and admins to provide feedback on vehicle conditions or report any issues with trips, ensuring that maintenance is carried out promptly to keep vehicles in good working order.

# CHAPTER 2

# SYSTEM STUDY

# 

### INTRODUCTION

AJCETRIPS is a mobile application designed specifically to manage and streamline the operations of the College Vehicle Management System. Developed as a Flutter Android app, the platform automates key tasks such as vehicle management, trip logging, maintenance scheduling, and real-time notifications. The app is tailored to simplify the operational process for college staff, making it more efficient and less time-consuming.

The system aims to enhance vehicle management by providing college drivers and administrators with a centralized tool for tracking vehicle details like insurance information, fitness test dates, and maintenance schedules. By integrating real-time notifications and trip management, AJCETRIPS ensures timely actions and better resource utilization.

This document will provide an in-depth overview of the AJCETRIPS project, its purpose, key features, user roles, and the technologies employed to develop the application. It serves as a guide for understanding the platform's functionality and the benefits it brings to the management of college transportation services.

### EXISTING SYSTEM

Currently, college vehicle management relies heavily on manual processes and traditional methods, which can lead to inefficiencies and lack of real-time information. In many institutions, drivers often manage vehicle details and trip information using paper records or basic spreadsheets, which are prone to errors and can be difficult to update.

When a driver needs to access vehicle information—such as insurance status, maintenance schedules, or upcoming trips—they may have to rely on physical documentation or outdated records, which can delay critical decision-making. Additionally, administrators often lack a centralized view of vehicle conditions and trip data, making it challenging to monitor usage, track maintenance, and ensure compliance with regulations.

Furthermore, notifications regarding important vehicle updates, such as insurance expirations or fitness test dates, are typically communicated through informal channels, leading to the risk of missed deadlines or overlooked maintenance needs. This reliance on outdated methods not only consumes time and resources but also hinders effective communication between drivers and administrators.

Overall, the existing system lacks automation and real-time tracking, resulting in inefficient vehicle management and increased potential for operational disruptions. AJCETRIPS aims to address these challenges by providing a streamlined, automated solution that enhances the management of college transportation services.

**2.2.1 NATURAL SYSTEM STUDIED**

The natural system of college vehicle management begins with identifying transportation needs within the institution. This process often involves faculty and staff communicating their requirements for vehicles to the administration, typically through informal channels or direct requests.

Once a request for transportation is made, the administration assesses the available vehicles, considering factors such as current maintenance status, insurance validity, and driver availability. This information is usually stored in physical records or basic spreadsheets, leading to difficulties in tracking vehicle conditions and scheduling trips.

As trips are planned, drivers manually record trip details, such as destinations and timings, in logbooks or spreadsheets. This system relies heavily on human intervention, which can result in potential errors in data entry and communication lapses. Furthermore, there is often no centralized system to monitor vehicle usage, making it challenging for administrators to ensure optimal resource allocation.

Overall, the existing natural system of vehicle management is characterized by its reliance on manual processes, which can lead to inefficiencies and operational challenges. The AJCETRIPS platform aims to transform this system by introducing automation and real-time tracking, enhancing the overall management of college transportation services.

**2.2.2 DESIGNED SYSTEM STUDIED**

The designed system study would involve analyzing the proposed system design and evaluating its effectiveness in addressing the pain points identified during the natural system study.

The following aspects would be studied in the designed system:

* System Architecture: The proposed system architecture would be studied to evaluate its scalability, reliability, and security. The designed system would need to be robust and able to handle large volumes of data, while also ensuring the security of sensitive customer and business data.
* User Interface Design: The user interface design would be evaluated to ensure that it is user-friendly, easy to navigate, and aesthetically pleasing. The designed system would need to be intuitive, with clear labels and instructions, and provide a consistent user experience across all modules.
* Functionality: The functionality of the designed system would be evaluated to ensure that it meets the requirements identified during the natural system study. The system would need to provide effective inventory management, sales management, and customer management capabilities, as well as real-time reporting and analysis capabilities.
* Testing: The designed system would be tested to ensure that it is free of bugs and errors, and that it performs as expected. The testing would include unit testing, integration testing, and system testing, as well as user acceptance testing to ensure that the system meets the requirements of the laptop shop owner.
* Maintenance and Support: The maintenance and support requirements of the designed system would be evaluated to ensure that the laptop shop owner has access to ongoing technical support and that the system can be easily maintained and upgraded as needed.

Based on the designed system study, any necessary modifications or improvements would be made to the system design to ensure that it meets the requirements of the travelers and effectively addresses the identified pain points. The designed system would then be implemented and deployed, with ongoing maintenance and support provided to ensure its continued effectiveness and reliability.

### DRAWBACKS OF EXISTING SYSTEM

* Limited integration
* Limited scalability
* Limited reporting capabilities
* High costs
* Complexity
* Long time interval to publish news

### PROPOSED SYSTEM

The AJCETRIPS system is a proposed solution designed to enhance the management of college vehicle operations through a dedicated Flutter Android App. This app aims to streamline processes for drivers and administrators by automating key tasks, ensuring real-time access to essential information, and improving overall efficiency. The application will feature a centralized dashboard for both drivers and administrators, providing easy access to vehicle details, trip logs, maintenance schedules, and notifications. It will integrate Firebase Cloud Messaging to send real-time alerts about important updates, such as insurance expirations, fitness test dates, and maintenance schedules, ensuring users are always informed and can take timely actions.

Additionally, AJCETRIPS will facilitate efficient trip management, allowing drivers to log trip details and track vehicle usage while giving administrators visibility over all trips to optimize resource allocation and minimize scheduling conflicts. The system will enable maintenance tracking, allowing users to monitor schedules and receive reminders for upcoming tasks, ensuring vehicles remain compliant and in good condition. A secure authentication mechanism will ensure appropriate access control based on user roles (admin or driver), enhancing data security. With integrated analytics and reporting features, administrators will be able to generate reports on vehicle utilization and maintenance history, supporting informed decision-making. Finally, as a mobile application, AJCETRIPS will be accessible from various Android devices, providing users with the flexibility and convenience to manage vehicle operations on the go. By addressing the limitations of the existing system, the proposed AJCETRIPS platform aims to transform college vehicle management into a more efficient, organized, and user-friendly process, ultimately enhancing the operational efficiency of college transportation services.

### ADVANTAGES OF PROPOSED SYSTEM

* Increased efficiency
* Better customer experience
* Improved management of news
* Enhanced reporting and analytics
* Better content accessibility
* Real-time updates: The proposed system would provide users with real-time updates on breaking news stories and events as they happen, ensuring users are always up-to-date with the latest information.

# CHAPTER 3

# REQUIREMENT ANALYSIS

## FEASIBILITY STUDY

Feasibility analysis is a crucial step in the planning of a project that involves determining the viability and suitability of the suggested project. It entails assessing the project's technical, economic, operational, legal, and scheduling aspects to determine whether it is feasible to start and finish the project successfully.

An online news portal offers many advantages over traditional print media, such as 24/7

Accessibility, the ability to incorporate multimedia elements like images, videos and link

### Economic Feasibility

Cost and benefit analysis are required to support the emerging system. Certain criteria must be placed to make sure that focus is placed on the project that will yield the best result at earliest.

The cost of developing and ongoing operations is considered. Also, the ability to generate revenue through advertising, subscription and other means.

Overall, we have estimated overall we are estimated that the benefits of organization are going to receive from the purposed system will surely overcome the initial cost and later running cost.

### Technical Feasibility

The technology used for website development should be established and stable. The website should also be compatible with common web browsers and mobile devices. There are several existing technologies and software systems that could be leveraged to develop the platform, including car rental management software, transportation management software, and payment processing systems. The cost of developing and implementing these systems will need to be carefully considered. The technology used for website development should be established and stable. The website should also be compatible with common web browsers and mobile devices. There are several existing technologies and software systems that could be leveraged to develop the platform. The project seems to have minimal limitations and has the potential to be implemented successfully. In the development process, the front-end of the system is built using PHP programming language, while the back-end relies on a MySQL server. To power the system, a robust Intel i3 core processor is utilized, along with 4GB of RAM and a spacious 1TB hard drive

### Behavioral Feasibility

The level to which the proposed system is accepted and simple to use by its intended users, such as readers and reporters, is referred to as the behavioral. It is a crucial component of a feasibility analysis because user adoption and usage are crucial to the system's success. Because it would accomplish the objectives after being developed and put into action, the

project would be advantageous. After carefully assessing all behavioral considerations, it is

determined that the project is behaviorally feasible.

### Questionnaire

1. What is the primary purpose of the AJCETRIPS system?

The primary purpose of AJCETRIPS is to enhance the management of college vehicle operations through an automated, user-friendly mobile application.

1. Who are the main users of the AJCETRIPS application?

The main users are college staff, including drivers and administrators, who manage vehicle details, trip schedules, and maintenance.

1. What are the key features of the AJCETRIPS vehicle management system?

Key features include real-time vehicle tracking, automated notifications for maintenance and compliance, trip management tools, and a centralized dashboard for easy access to information.

1. How does AJCETRIPS improve communication between drivers and administrators?

AJCETRIPS improves communication through real-time notifications and a centralized platform that allows both parties to access and update information easily.

1. What technology stack is used to develop the AJCETRIPS application?

The front end is developed using the Flutter framework for mobile accessibility, and the back end utilizes Firebase for real-time data management.

1. How does the AJCETRIPS system ensure data security?

Data security is ensured through secure authentication mechanisms and role-based access control to protect sensitive information.

1. What types of notifications can users expect from the AJCETRIPS system?

Users can expect notifications for maintenance schedules, insurance renewals, trip confirmations, and important updates related to vehicle management.

1. How does the system facilitate better resource management for the college?

AJCETRIPS allows administrators to monitor vehicle usage, maintenance history, and trip details in real-time, enabling more efficient resource allocation and planning.

1. What challenges does the AJCETRIPS project aim to address compared to existing systems?

The project aims to address challenges such as manual record-keeping, lack of real-time data, and inefficient communication, which lead to delays and errors in vehicle management.

1. How does the AJCETRIPS system enhance the overall user experience for college staff?

By providing a user-friendly interface, automated features, and centralized access to critical information, AJCETRIPS significantly simplifies the vehicle management process for college staff.

## SYSTEM SPECIFICATION

### Hardware Specification

Processor - intel core i3

RAM - minimum 2gb

Hard disk - minimum 256 GB

### Software Specification

Front End - Flutter

Backend - MYSQL, Firebase

Client on PC - Windows 7 and above.

Technologies used - Dart, Local Notifications, Firebase,ML

The AJCETRIPS project leverages Flutter, a UI toolkit that enables the development of natively compiled applications for mobile, web, and desktop from a single codebase. Flutter offers a rich set of pre-built widgets, facilitating high-performance mobile app development. User authentication is streamlined through Firebase Authentication, which provides secure login and registration options via email/password and social media accounts. The application is built using Dart, a programming language optimized for creating mobile, desktop, server, and web applications, known for its ease of learning and strong support for asynchronous programming. To enhance user engagement, the system incorporates Local Notifications, allowing for the delivery of alerts and reminders directly to users’ mobile devices, such as notifications regarding maintenance schedules, trip confirmations, and other important updates related to vehicle management.

## SOFTWARE DESCRIPTION

### Flutter

Flutter is an open-source UI toolkit developed by Google, specifically designed for building natively compiled applications for mobile, web, and desktop from a single codebase. It allows developers to create high-performance applications with a visually appealing and responsive user interface. Flutter uses a reactive programming model, enabling quick updates and seamless user experiences. Its rich set of pre-built widgets and customizable components make it ideal for creating user-friendly applications tailored to specific needs. By utilizing the Dart programming language, Flutter applications benefit from fast development cycles and strong performance due to Dart's asynchronous programming capabilities.

### Firebase Authentication

Firebase Authentication is a service provided by Google that simplifies the process of authenticating users across various platforms. It supports a range of authentication methods, including email/password, phone authentication, and integration with social media accounts like Google and Facebook. This flexibility allows developers to implement secure and user-friendly authentication solutions in their applications. Firebase Authentication also provides a robust backend infrastructure that manages user sessions, ensuring that only authorized users can access sensitive data. With its easy integration into mobile applications, Firebase Authentication enhances user experience while maintaining high security standards.

### Local Notifications

Local Notifications are a crucial feature in mobile applications, enabling developers to send alerts and reminders directly to users' devices. This functionality is vital for applications like AJCETRIPS, where timely notifications regarding maintenance schedules, trip confirmations, and important updates are essential for efficient vehicle management. Local notifications can be scheduled to appear at specific times or triggered by certain events within the application, ensuring users are always informed and engaged. By utilizing local notifications, AJCETRIPS enhances its communication capabilities, allowing staff to manage their tasks effectively and remain up-to-date on critical information.

# CHAPTER 4

# SYSTEM DESIGN

* 1. **INTRODUCTION**

System design is the process of creating a system's architecture, parts, and interfaces to ensure that it satisfies the needs of its users. It is an interdisciplinary engineering activity that makes successful systems possible. A system can be defined as a cohesive group of parts working together to achieve a specific goal. Systems design involves defining the software and hardware architecture, as well as its components, modules, interfaces, and data, in order to make a system capable of achieving a set of clearly defined operational requirements.

Early in the development cycle, systems design focuses on defining customer needs and necessary functionality, documenting requirements, then moving on to design synthesis and system validation while taking into account the larger issue, which includes:

* Operations
* Performance
* Test and integration
* Manufacturing
* Cost and schedule
* Deployment
* Training and support
* Maintenance
* Disposal

## UML DIAGRAM

The Unified Modelling Language (UML) is a standardized notation system used for modeling software systems. The UML diagram is a graphical representation of the system design and is an essential part of the system design phase.

In the UML diagram, the various components and functionalities of the proposed web-based application for currency traders are represented using various types of diagrams, such as use case diagrams, class diagrams, activity diagrams, and sequence diagrams. These diagrams help in visualizing the system's architecture, behaviour, and interactions between different components.

Use case diagrams are used to identify the various types of users who will interact with the platform and the actions they will perform. Class diagrams are used to model the data and the relationships between different classes. Activity diagrams are used to model the workflow and the sequence of actions that need to be performed to complete a specific task. Sequence diagrams are used to model the interactions between different components of the system.

## USE CASE DIAGRAM

The use case diagram is a type of UML diagram that is used to identify the various actors or users of the system and the actions or tasks that they will perform.

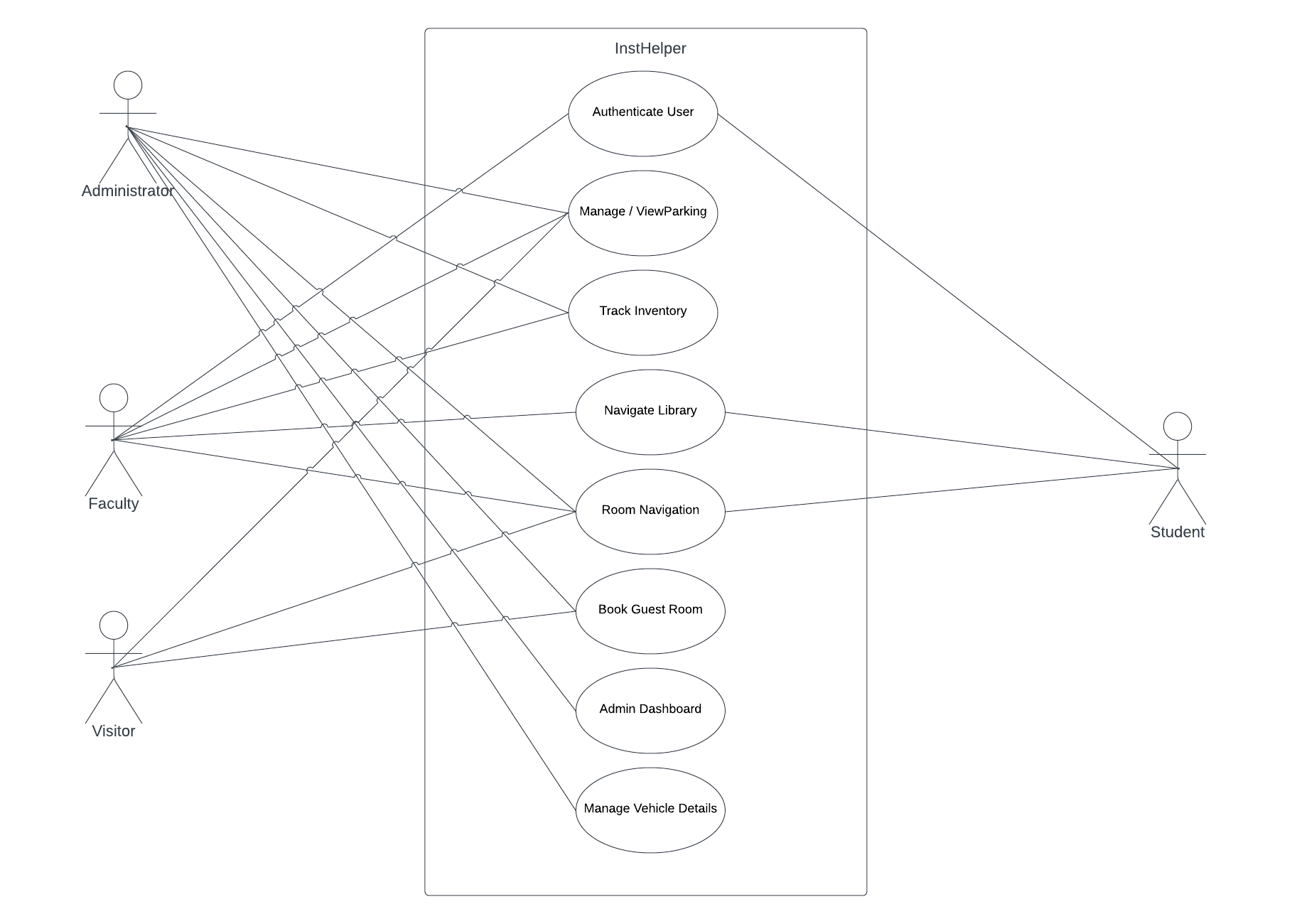
****

Fig 1: use case diagram

## SEQUENCE DIAGRAM

## The sequence diagram is a type of UML diagram that is used to model the interactions between the various components of the system. In the case of the proposed web-based application for currency traders, the sequence diagram is used to model the interactions between the users and the system's components, such as the database, the analysis engine, and the reporting engine.

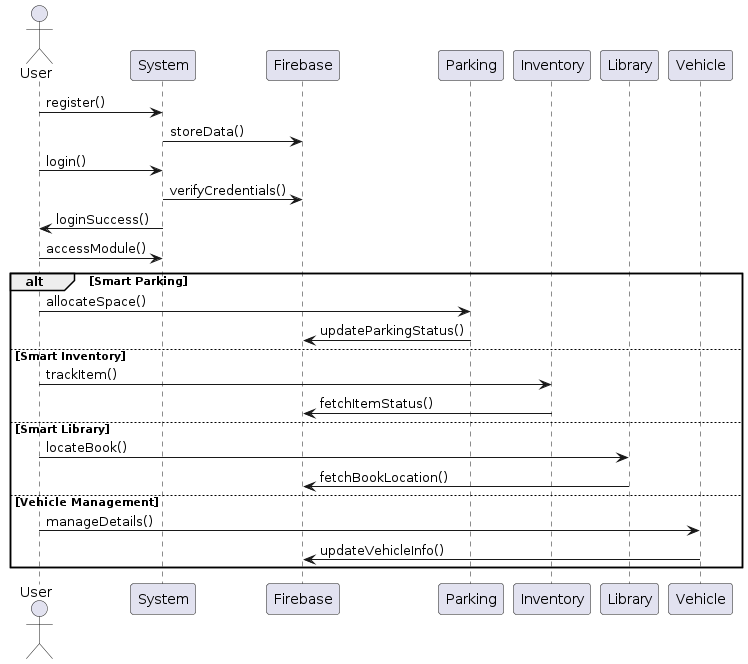
****

Fig 2: sequence diagram

## 4.2.2 State Chart Diagram

A state diagram, also known as a state machine diagram or state chart diagram, is an

illustration of the states an object can attain as well as the transitions between those

states in the Unified Modeling Language (UML). A state diagram, also known as a

state machine diagram or state chart diagram, is an illustration of the states an object

can attain as well as the transitions between those states in the Unified Modeling

Language (UML).

## Activity Diagram

An activity diagram is a flowchart that outlines all the activities performed by a system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity.

It shows everything from start to finish, defining the various decision paths and steps that need to happen to move from one activity to the next. The steps can be chronological, branched, or simultaneous. It allows business analysts to effectively plot and manage various workflows in one location.

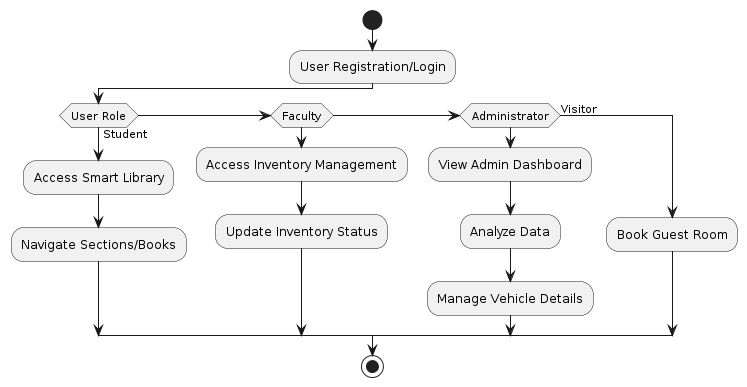
****

Fig 3: activity diagram

## Class Diagram

A class diagram allows you to map the structures of a system displaying various classes, attributes, operations, and relationships between objects. Both software engineers and business managers use this interaction diagram to model different connections involved within a process.

In the diagram, class is represented with a rectangle. Each rectangle is split vertically into three sections. The top section has the name of the class. The second and third sections provide details about class operations, behaviors, and attributes.

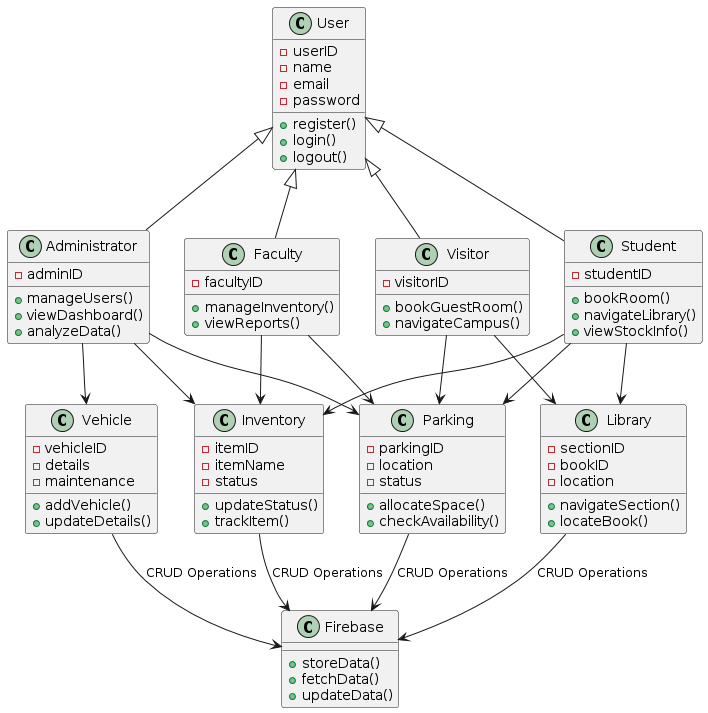
****

Fig 4: class diagram

## Object Diagram

Object diagrams show the attributes of each object in a system and how those objects relate to

one another at a specific time. Object diagrams like this are often used alongside class diagrams. Because an object diagram tends to represent a certain part of a class diagram.

In this situation, a class diagram would illustrate the entire baking system. An object diagram

will focus on the use of different objects throughout the system. This helps you verify the effectiveness of your class diagram. A UML object diagram represents a specific instance of a class diagram at a certain moment in time. When represented visually, you'll see many

similarities to the class diagram. An object diagram focuses on the attributes of a set of objects and how those objects relate to each other. In UML, object diagrams provide a snapshot of the instances in a system and the relationships between the instances. By instantiating the model elements in a class diagram, you can explore the behavior of a system at a point in time.

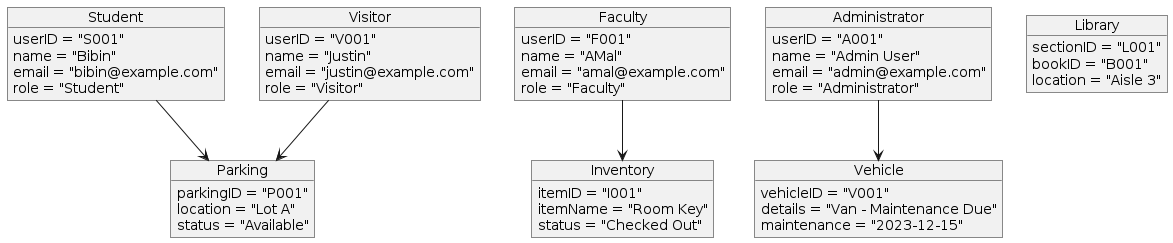
****

Fig 5: object diagram

## Component Diagram

Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system. A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

**4.2.8 Deployment Diagram**

Deployment diagrams show the configuration of processing nodes, as well as all the

components that live on them. In other words, they show how software runs on components within a system. It helps users model the physical aspects of object-oriented software.

These diagrams are mostly used by system engineers. By using this diagram, they can keep

track of their entire hardware mesh and prepare the system for launch without any issues. It also

allows them to see performance, maintainability, and scalability. A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

Deployment diagrams are typically used to visualize the physical hardware and software of a system.

**4.2.9 Collaboration Diagram**

Collaboration diagrams are used to show the relationships and interactions between different components or objects in a system, and how they communicate with each other to achieve a specific goal. They can be used to visualize the flow of data and control between different components or objects, and to identify potential bottlenecks or areas of inefficiency in a system.

Collaboration diagrams can be used to model complex systems and to help identify potential issues or areas of improvement. They can also be used to communicate the design of a system to other stakeholders, such as developers, project managers, or clients.

## 4.3 USER INTERFACE DESIGN USING FIGMA

**Form Name: Login**

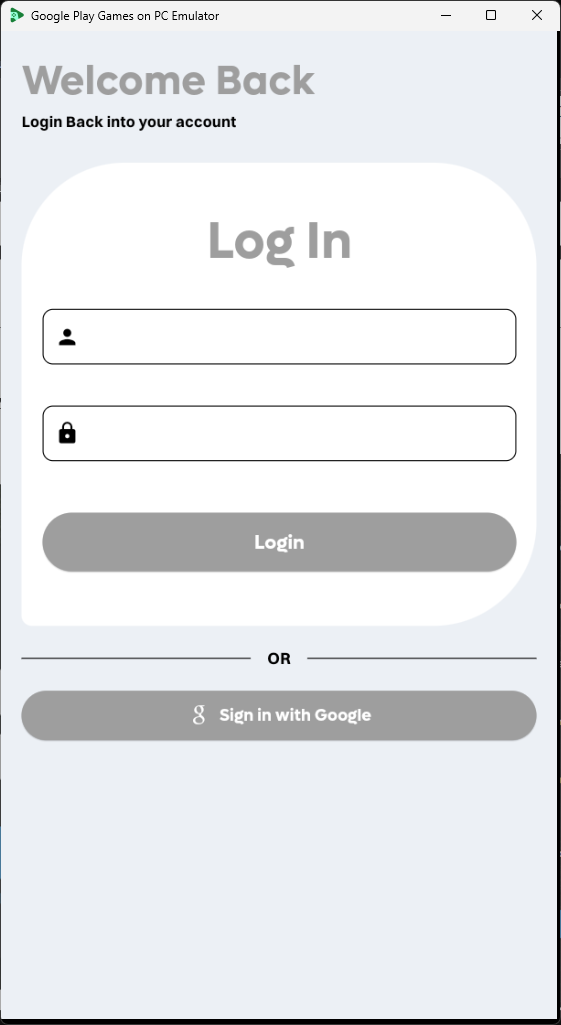


Fig 6: login

**Form Name: Home**

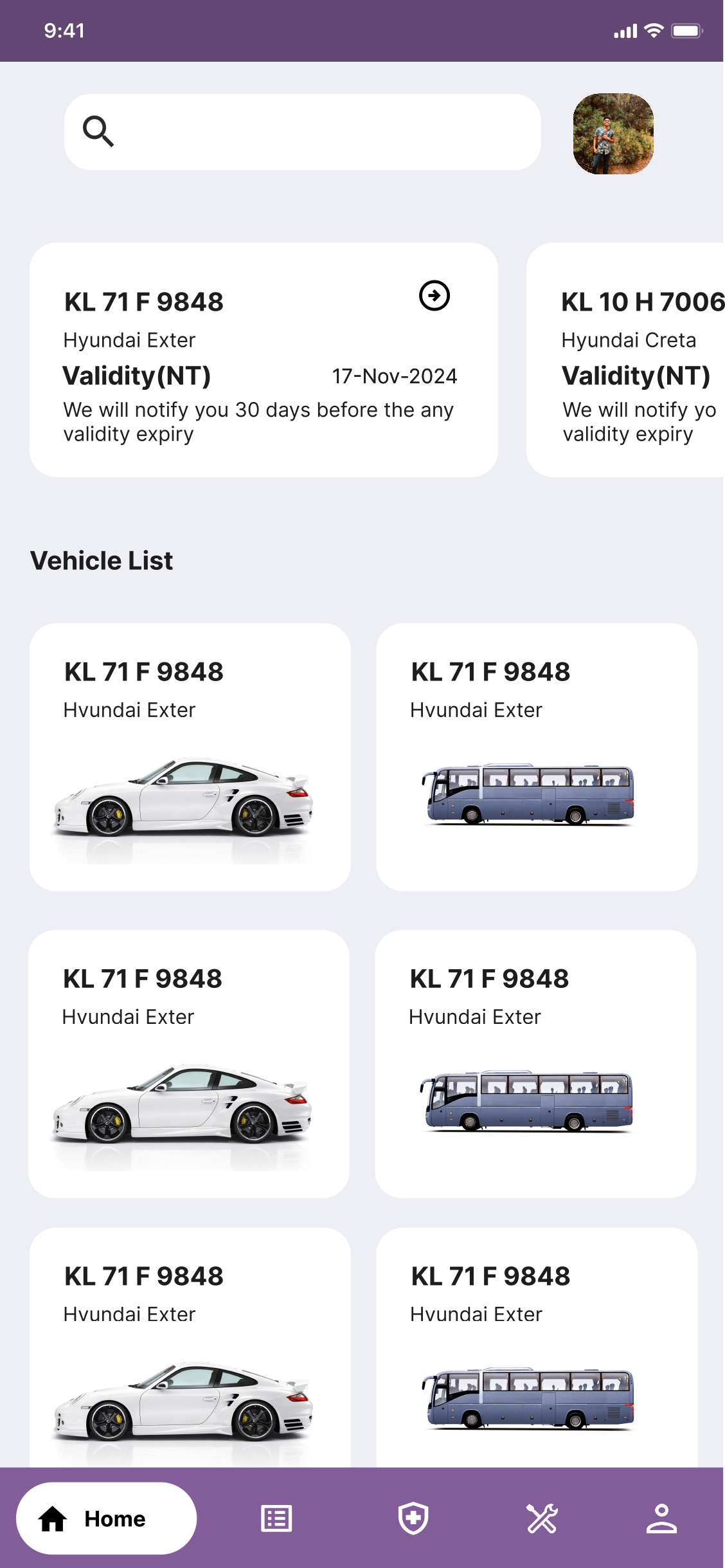


Fig 7: Home Page

**Form Name: Vehicle List**

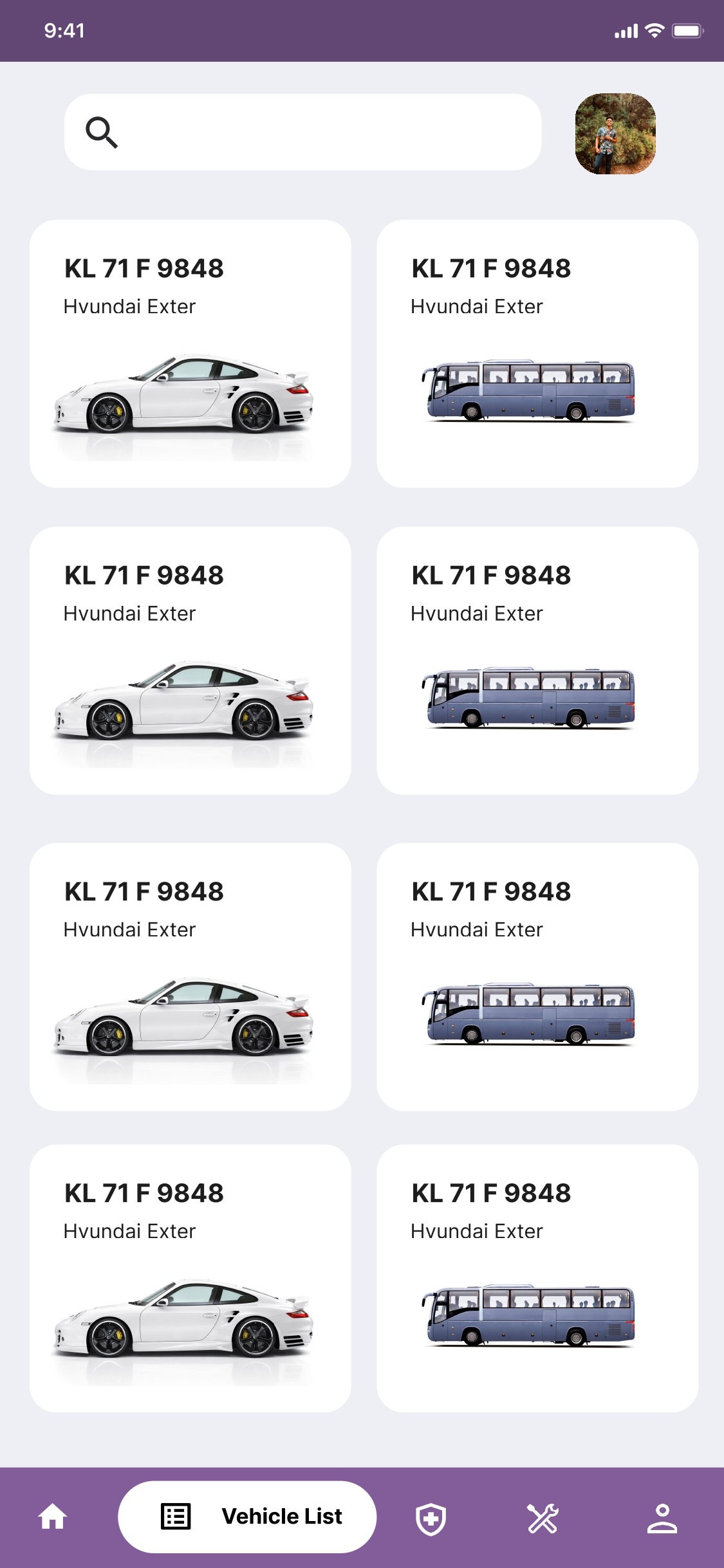


Fig 7: Vehicle List

## 4.4 DATABASE DESIGN

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database

management system (DBMS). Together, the data and the DBMS, along with the applications

that are associated with them, are referred to as a database system, often shortened to just database. Data within the most common types of databases in operation today is typically

modeled in rows and columns in a series of tables to make processing and data querying

efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data. SQL is a programming language used by nearly all relational databases to query, manipulate,

and define data, and to provide access control. SQL was first developed at IBM in the 1970s

with Oracle as a major contributor, which led to implementation of the SQL ANSI standard,

SQL has spurred many extensions from companies such as IBM, Oracle, and Microsoft.

Although SQL is still widely used today, new programming languages are beginning to appear.

### 4.4.1 Relational Database Management System (RDBMS)

A relational database management system (RDBMS) is a collection of programs and

capabilities that enable IT teams and others to create, update, administer and otherwise

interact with a relational database. dbases store data in the form of tables, with most

commercial relational database management systems using Structured Query

Language (SQL) to access the database. However, since SQL was invented after the

initial development of the relational model, it is not necessary for RDBMS use.

The RDBMS is the most popular database system among organizations across the

world. It provides a dependable method of storing and retrieving large amounts of data

while offering a combination of system performance and ease of implementation.

Other differences between database management systems and relational database

management systems include:

• Number of allowed users. While a DBMS can only accept one user at a time, an

RDBMS can operate with multiple users.

• Hardware and software requirements. A DBMS needs less software and hardware

than an RDBMS.

• Amount of data. dbases can handle any amount of data, from small to large, while a

DBMS can only manage small amounts.

### 4.4.2 Normalization

Normalization is the process to eliminate data redundancy and enhance data integrity

in the table. Normalization also helps to organize the data in the database. It is a

multistep process that sets the data into tabular form and removes the duplicated data

from the relational tables. Normalization is a process of decomposing the relations

into relations with fewer attributes.

Normalization is used to minimize the redundancy from a relation or set of relations.

It is also used to eliminate undesirable characteristics like Insertion, Update, and

Deletion Anomalies.

Data modification anomalies can be categorized into three types:

• Insertion Anomaly: Insertion Anomaly refers to when one cannot insert a new tuple

into a relationship due to lack of data.

• Deletion Anomaly: The delete anomaly refers to the situation where the deletion of

data results in the unintended loss of some other important data.

• Updation Anomaly: The update anomaly is when an update of a single data value

requires multiple rows of data to be updated.

Normalization works through a series of stages called Normal forms. The normal

forms apply to individual relations. The relation is said to be in particular normal form

if it satisfies constraints.

1) First Normal Form: A relation is in 1NF if it contains an atomic value.

2) Second Normal Form: A relation will be in 2NF if it is in 1NF and all non-key

attributes are fully functional dependent on the primary key.

3) Third Normal Form: A relation will be in 3NF if it is in 2NF and no transition

dependency exists.

4) Boyce Codd Normal Form: A stronger definition of 3NF is known as Boyce Codd's

normal form.

5) Fourth Normal Form: A relation will be in 4NF if it is in Boyce Codd's normal form

and has no multi-valued dependency.

6) Fifth Normal Form: A relation is in 5NF. If it is in 4NF and does not contain any

join dependency, joining should be lossless.

### 4.4.3 Sanitization

Data sanitization involves purposely, permanently deleting, or destroying data from

a storage device, to ensure it cannot be recovered. The most common scenario for

data sanitization is re-imaging. This usually happens when equipment is reassigned

to new users. Imaging overwrites the core operating system files, file allocation table

(FAT), etc. However, the old data is not actually deleted.

Ordinarily, when data is deleted from storage media, the media is not really erased

and can be recovered by an attacker who gains access to the device. This raise

serious concerns for security and data privacy. With sanitization, storage media is

cleansed so there is no leftover data on the device, and no data can be recovered,

even with advanced forensic tools.

There are four primary methods to achieve data sanitization: physical destruction,

data erasure, cryptographic erasure, and data masking.

**4.4.4 Indexing**

Indexing refers to a data structure technique that is used for quickly retrieving entries

from database files using some attributes that have been indexed. In database

systems, indexing is comparable to indexing in books. The indexing attributes are

used to define the indexing.

o Indexing is used to optimize the performance of a database by minimizing the

number of disk accesses required when a query is processed.

o The index is a type of data structure. It is used to locate and access the data in a

database table quickly.

Indexing in DBMS is of the following types:

• Ordered Index

• Primary Index

• Clustering Index

• Sparsing Index

### 4.5 TABLE DESIGN

**1.** **tbl\_drivers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Fieldname** | **Datatype** | **Key** | **Constraints** | **Description** |
| 1 | id | int(11) | PRIMARY | NOT NULL, AUTO\_INCREMENT | Unique identifier for each driver |
| 2 | name | varchar(30) |  |  | Driver's name |
| 3 | contact | varchar(30) |  |  | Driver's contact number |
| 4 | license | varchar(100) |  |  | Driver's license number |
| 5 | onCreated | timestamp |  | NOT NULL, DEFAULT current\_timestamp() | Timestamp of record creation |

### 2. tbl\_fitness

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Fieldname** | **Datatype** | **Key** | **Constraints** | **Description** |
| 1 | id | int(11) | PRIMARY | NOT NULL, AUTO\_INCREMENT | Unique identifier for each fitness record |
| 2 | vehicle \_id | varchar(30) | Foreign | NOT NULL, FOREIGN KEY (vehicle\_id) REFERENCES tbl\_vehicles(vehicle\_id) | Vehicle registration number |
| 3 | exp\_date | date |  |  | Expiration date of fitness certificate |
| 4 | documents | varchar(300) |  |  | URL to fitness documents |
| 5 | onCreated | timestamp |  | NOT NULL, DEFAULT current\_timestamp() | Timestamp of record creation |

### 3. tbl\_fuel

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Fieldname** | **Datatype** | **Key** | **Constraints** | **Description** |
| 1 | fuel\_id | int(11) | PRIMARY | NOT NULL, AUTO\_INCREMENT | Unique identifier for each fuel type |
| 2 | type | varchar(20) |  |  | Name of the fuel type |

### 4. tbl\_insurance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Fieldname** | **Datatype** | **Key** | **Constraints** | **Description** |
| 1 | id | int(11) | PRIMARY | NOT NULL, AUTO\_INCREMENT | Unique identifier for each insurance record |
| 2 | vehicle\_id | varchar(30) | Foreign | NOT NULL, FOREIGN KEY (vehicle\_id) REFERENCES tbl\_vehicles(vehicle\_id) | Vehicle registration number |
| 3 | exp\_date | date |  |  | Expiration date of insurance |
| 4 | documents | varchar(300) |  |  | URL to insurance documents |
| 5 | onCreated | timestamp |  | NOT NULL, DEFAULT current\_timestamp() | Timestamp of record creation |

### 5. tbl\_pollution

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Fieldname** | **Datatype** | **Key** | **Constraints** | **Description** |
| 1 | id | int(11) | PRIMARY | NOT NULL, AUTO\_INCREMENT | Unique identifier for each pollution record |
| 2 | vehicle\_id | varchar(30) | Foreign | NOT NULL, FOREIGN KEY (vehicle\_id) REFERENCES tbl\_vehicles(vehicle\_id) | Vehicle registration number |
| 3 | exp\_date | date |  |  | Expiration date of pollution certificate |
| 4 | documents | varchar(300) |  |  | URL to pollution documents |
| 5 | onCreated | timestamp |  | NOT NULL, DEFAULT current\_timestamp() | Timestamp of record creation |

### 6. tbl\_trips

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Fieldname** | **Datatype** | **Key** | **Constraints** | **Description** |
| 1 | id | int(11) | PRIMARY | NOT NULL, AUTO\_INCREMENT | Unique identifier for each trip |
| 2 | vehicle\_id | varchar(300) | Foreign | NOT NULL, FOREIGN KEY (vehicle\_id) REFERENCES tbl\_vehicles(vehicle\_id) | JSON array of vehicle registration numbers |
| 3 | driver | varchar(300) |  |  | JSON array of driver names |
| 4 | purpose | varchar(300) |  |  | Purpose of the trip |
| 5 | starting\_time | varchar(10) |  |  | Starting time of the trip |
| 6 | route | varchar(300) |  |  | JSON array of route coordinates or place names |
| 7 | starting\_km | varchar(300) |  |  | JSON array of starting kilometers for each vehicle |
| 8 | ending\_km | varchar(300) |  |  | JSON array of ending kilometers for each vehicle |
| 9 | onCreated | timestamp |  | NOT NULL, DEFAULT current\_timestamp() | Timestamp of record creation |
| 10 | onUpdate | timestamp |  | NOT NULL, DEFAULT current\_timestamp() ON UPDATE current\_timestamp() | Timestamp of last update |

### 7. tbl\_vehicle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Fieldname** | **Datatype** | **Key** | **Constraints** | **Description** |
| 1 | id | int(11) | PRIMARY | NOT NULL, AUTO\_INCREMENT | Unique identifier for each vehicle |
| 2 | assigned\_driver | varchar(50) | Foreign | NOT NULL, FOREIGN KEY (driver\_id) REFERENCES tbl\_drivers(driver\_id) | Name of the assigned driver |
| 3 | chassis\_no | varchar(100) |  |  | Chassis number of the vehicle |
| 4 | current\_mileage | double |  |  | Current mileage of the vehicle |
| 5 | total\_km | double |  |  | Total kilometers traveled |
| 6 | emergency\_contact | varchar(10) |  |  | Emergency contact number |
| 7 | engine\_no | varchar(100) |  |  | Engine number of the vehicle |
| 8 | fuel\_type | varchar(10) | Foreign | NOT NULL, FOREIGN KEY (fuel\_type) REFERENCES tbl\_fuels(fuel\_id) | Type of fuel used |
| 9 | model | varchar(100) |  |  | Model of the vehicle |
| 10 | ownership | varchar(30) |  |  | Owner of the vehicle |
| 11 | purpose\_of\_use | varchar(100) |  |  | Purpose for which the vehicle is used |
| 12 | registration\_date | varchar(20) |  |  | Date of vehicle registration |
| 13 | registration\_number | varchar(30) |  |  | Vehicle registration number |
| 14 | uploaded\_files | text |  |  | JSON array of URLs to uploaded files |
| 15 | vehicle\_type | varchar(20) | Foreign | NOT NULL, FOREIGN KEY (vehicle\_type) REFERENCES tbl\_vehicle\_type(id) | Type of vehicle |
| 16 | onCreated | timestamp |  | NOT NULL, DEFAULT current\_timestamp() | Timestamp of record creation |
| 17 | onUpdated | timestamp |  | NOT NULL, DEFAULT current\_timestamp() ON UPDATE current\_timestamp() | Timestamp of last update |

### 8. tbl\_vehicle\_gallery

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Fieldname** | **Datatype** | **Key** | **Constraints** | **Description** |
| 1 | id | int(11) | PRIMARY | NOT NULL, AUTO\_INCREMENT | Unique identifier for each gallery entry |
| 2 | vehicle\_id | varchar(30) | Foreign | NOT NULL, FOREIGN KEY (vehicle\_id) REFERENCES tbl\_vehicles(vehicle\_id) | Vehicle registration number |
| 3 | image | varchar(500) |  | NOT NULL | JSON array of image URLs |

### 9. tbl\_vehicle\_type

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Fieldname** | **Datatype** | **Key** | **Constraints** | **Description** |
| 1 | id | int(11) | PRIMARY | NOT NULL, AUTO\_INCREMENT | Unique identifier for each vehicle type |
| 2 | type | varchar(30) |  |  | Name of the vehicle type |
| 3 | image | varchar(300) |  |  | URL to the image representing the vehicle type |

# CHAPTER 5

# SYSTEM TESTING

* 1. **INTRODUCTION**

The System Testing phase is a critical component of the software development life cycle, which aims to ensure that the software system meets the specified requirements and performs as expected. System testing involves testing the entire system as a whole, including all of its components and sub-systems. It is a comprehensive process that involves various types of testing, such as functional testing, performance testing, security testing, and usability testing, among others.

The purpose of this section of the test report is to outline the results of the system testing phase of the project. The testing was conducted to ensure that the software system meets the functional and non-functional requirements specified in the project specifications. This report includes a description of the testing approach, the testing environment, and the test results. Additionally, it presents any defects or issues identified during the testing process and outlines the steps taken to resolve them. Overall, this report provides an overview of the system testing phase, demonstrating how the system has been tested and the results of the testing.

## TEST PLAN

A test plan is a document that outlines the objectives, scope, approach, and resources required for testing a software system or application. The purpose of a test plan is to guide the testing process and ensure that all necessary tests are conducted and documented.

Here are some of the key components of a test plan:

• Test objectives: This section outlines the overall goals and objectives of the testing effort, and how the testing will help to achieve the project's overall goals.

• Test scope: This section defines the scope of the testing effort, including the areas of the system to be tested, the types of testing to be conducted, and any limitations or constraints on the testing effort.

• Test approach: This section outlines the overall approach to be taken for testing the system, including the testing methodology, tools and techniques to be used, and any specific testing strategies or tactics.

• Test schedule: This section provides a detailed timeline of the testing effort, including the start and end dates for each phase of testing, as well as any milestones or deadlines that must be met.

• Test resources: This section lists the resources required for the testing effort, including personnel, hardware, software, and any other equipment or facilities needed for testing.

• Test deliverables: This section lists the deliverables that will be produced during the testing effort, such as test plans, test cases, test reports, and any other documentation.

• Test risks: This section identifies the potential risks and issues that may arise during the testing effort, and outlines strategies for mitigating or addressing them.

Overall, a test plan is an essential tool for ensuring that a software system is thoroughly tested and meets the desired quality standards. It provides a roadmap for the testing effort, ensuring that all necessary tests are conducted and that the results are properly documented and reported.

### Unit Testing

Unit testing is a type of software testing that involves testing individual units or components of a software application in isolation from the rest of the system. The purpose of unit testing is to ensure that each unit of code, such as a function or method, works correctly and meets its intended functionality.

Unit tests are typically automated, meaning that they can be run quickly and easily whenever changes are made to the code. They are also repeatable, meaning that the same tests can be run over and over again to ensure that the code continues to function as expected.

Unit testing can help catch bugs early in the development cycle, before they have a chance to cause larger problems in the system. It can also help developers to identify and fix issues more quickly, leading to faster development times and higher quality software.

In order to perform unit testing, developers use specialized testing frameworks and tools that help automate the process and provide feedback on the success or failure of individual tests. Common unit testing frameworks include JUnit for Java, NUnit for .NET, and pytest for Python, among others.

### Integration Testing

Integration testing is a type of software testing that aims to test how different components or modules of a software application work together. The purpose of integration testing is to verify that these components can communicate and exchange data with each other as expected, and that they function correctly as a unified system.

Integration testing typically occurs after unit testing has been completed, and it may involve testing different modules or components of the application together in pairs or in larger groups. This can be done in a variety of ways, including using stubs or mock objects to simulate the behavior of missing components, or by testing against a live system or database.

There are several different types of integration testing, including:

* Big Bang Integration: In this approach, all components are integrated at once and tested as a whole.
* Top-Down Integration: In this approach, testing starts from the top layer of the application and works its way down, integrating each layer as it goes.
* Bottom-Up Integration: In this approach, testing starts from the bottom layer of the application and works its way up, integrating each layer as it goes.
* Sandwich Integration: In this approach, testing starts from the top layer of the application, works its way down to the bottom layer, and then back up to the top layer again.

Integration testing is important because it helps to ensure that the different components of an application work together as intended, and that any problems or issues are identified and fixed before the application is released to users.

### Validation Testing or System Testing

Validation testing, also known as system testing, is a type of software testing that aims to ensure that a software system meets its intended purpose and satisfies the requirements of the stakeholders. It is typically performed after integration testing has been completed and involves testing the entire system as a whole, rather than individual components.

The purpose of validation testing is to verify that the system functions correctly in the context of the larger environment in which it will be used. This includes testing the system's functionality, usability, performance, reliability, and security, among other factors.

Validation testing may involve a combination of manual and automated testing techniques, and may be conducted using various testing methods, such as:

* Functional testing: This involves testing the system's functionality to ensure that it meets the requirements of the stakeholders.
* Usability testing: This involves testing the system's ease of use and user interface to ensure that it is intuitive and user-friendly.
* Performance testing: This involves testing the system's speed, scalability, and resource usage to ensure that it can handle the expected workload.
* Security testing: This involves testing the system's security features and vulnerabilities to ensure that it is protected against unauthorized access or malicious attacks.
* Compatibility testing: This involves testing the system's compatibility with different platforms, devices, and software configurations.

Validation testing is important because it helps to ensure that the software system is ready for release and meets the expectations of the stakeholders. It can help to identify and address any issues or defects before the system is deployed, which can save time and resources in the long run.

### Output Testing or User Acceptance Testing

Output testing, also known as user acceptance testing (UAT), is a type of software testing that involves testing the software system from the perspective of the end-user. The purpose of user acceptance testing is to verify that the system meets the requirements and expectations of the stakeholders, and that it functions correctly in the context of the user's workflow and environment.

User acceptance testing is typically performed by a group of users or stakeholders who are representative of the target audience for the software system. These users are given specific scenarios or tasks to perform, and they provide feedback on their experience using the system. The focus of user acceptance testing is on the output or results of the system, rather than the individual components or technical details.

The main objectives of user acceptance testing are to:

1. Verify that the system meets the requirements and expectations of the stakeholders.
2. Validate that the system is easy to use and understand.
3. Confirm that the system is reliable, accurate, and produces the expected output.
4. Ensure that the system integrates seamlessly with other systems or tools used by the users.
5. Identify any remaining defects or issues that need to be addressed before the system is released.

User acceptance testing is important because it provides a final check to ensure that the software system is ready for deployment and meets the needs of the end-users. It helps to validate that the system will be accepted and adopted by the users, which can ultimately determine the success of the project.

* + 1. **Automation Testing**

Automation testing is a type of software testing that involves the use of tools and software to automate the execution of test cases and the comparison of actual results with expected results. The purpose of automation testing is to improve the efficiency and effectiveness of the testing process by reducing the time and effort required to execute tests and analyze results.

Automation testing can be applied to different types of testing, including unit testing, integration testing, and system testing. It involves the use of specialized tools and frameworks that can automate the testing process, such as:

* Test automation frameworks: These provide a structured approach to automate the testing process, including test case management, test data management, and reporting.
* Test scripting tools: These enable the creation and execution of automated test scripts, which can simulate user interactions with the software system.
* Test management tools: These enable the organization and scheduling of automated tests, and provide tools for tracking test results and defects.
* Automation testing offers several advantages over manual testing, including:
* Increased efficiency: Automation testing can execute test cases faster and more accurately than manual testing, which can save time and reduce the cost of testing.
* Improved test coverage: Automation testing can cover a larger number of test cases and scenarios than manual testing, which can help to identify defects and issues that may be missed by manual testing.
* Increased accuracy: Automation testing can eliminate human errors and biases that may occur in manual testing, which can improve the accuracy and reliability of test results.
* Reusability: Automated test scripts can be reused across different testing cycles and environments, which can save time and effort in the long run.

However, automation testing also has some limitations, such as the need for specialized skills and resources to develop and maintain automated tests, as well as the inability to test certain aspects of the system that require human intuition or judgment.

* + 1. **Selenium Testing**

Selenium testing is a popular open-source automation testing tool used for web application testing. It allows testers to automate the testing of web applications across different browsers and platforms, using various programming languages such as Java, Python, C#, and more.

Selenium testing supports different types of testing, including functional testing, regression testing, and compatibility testing. It offers several features that make it a popular choice for web application testing, such as:

* Cross-browser testing: Selenium testing allows for automated testing of web applications across different browsers, including Chrome, Firefox, Safari, and Internet Explorer.
* Record and playback: Selenium IDE, a Selenium testing tool, allows testers to record and playback interactions with the web application, which can be useful for creating test scripts.
* Multi-language support: Selenium testing supports various programming languages, which allows testers to write test scripts in a language of their choice.
* Integration with other tools: Selenium testing can be integrated with other testing tools, such as TestNG and JUnit, for better test management and reporting.
* Parallel testing: Selenium Grid, another Selenium testing tool, allows testers to execute tests across different browsers and platforms in parallel, which can save time and increase testing efficiency.

Selenium testing is widely used for web application testing, particularly for testing complex web applications with dynamic content and user interfaces. However, it also has some limitations, such as the need for regular updates to keep up with changes in browsers and web technologies, as well as the complexity of maintaining and scaling test scripts in large and complex projects.

**Example:**

**Test Case 1**

import time

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

# set up the driver

driver = webdriver.Chrome()

# navigate to the login page

driver.get("http://localhost/miniproject/login1.php")

driver.set\_window\_size(1536,816)

# enter the username

username\_input = driver.find\_element(By.ID, "name")

username\_input.send\_keys("ROHAN")

time.sleep(2)

# enter the password

password\_input = driver.find\_element(By.ID, "password")

password\_input.send\_keys("a12345678")

time.sleep(3)

# submit the form

submit\_button = driver.find\_element(By.ID, "submit")

submit\_button.click()

time.sleep(5)

welcome\_message=driver.find\_element(By.XPATH, "//a[contains(text(),'Home')]").text

if(welcome\_message=='Home'):

print("\n\033[32m\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n Testing success \n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

else:

print("\n\033[31m\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n Testing failed \n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

# close the browser window

driver.quit()

****

Fig 8: output

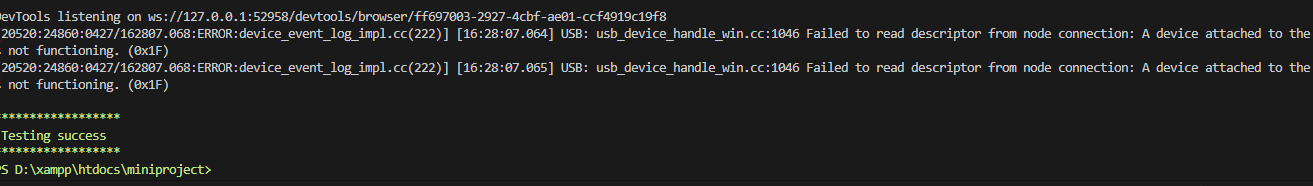


Fig 9: Result

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case 1** | | | | | |
| **Project Name: Newzilla** | | | | | |
| **Login Test Case** | | | | | |
| **Test Case ID: Test\_1** | | | **Test Designed By: Rohan** | | |
| **Test Priority(Low/Medium/High):** | | | **Test Designed Date: 24/04/2023** | | |
| **Module Name**: login | | | **Test Executed By: Rohan** | | |
| **Test Title : login of users** | | | **Test Execution Date: 24/04/2023** | | |
| **Description: login of users** | | |  | | |
| **Pre-Condition:** User has valid username and password | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status(Pass/**  **Fai l)** |
| 1 | Navigate to login page | http://localhost/miniproject/login1.php | Login page loads successfully | Login page loaded successfully | Pass |
| 2 | Fill in username and password fields | "Rohan" for username and "a12345678" for password | Username and password fields are successfully filled in | Username and password fields are successfully filled in | Pass |
| 3 | Find the "submit" button and click it | ID: "submit" | The button is clicked and successfully logged in | User is successfully logged in | Pass |
| 4 | Close the browser window | driver.quit() | Browser window is successfully closed | Browser window is successfully closed | Pass |
|  |  |  |  |  |
| **Post-Condition: Logged succesfully** | | | | | |

**Test Case 2:**

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

# Set the path of the ChromeDriver executable

driver\_path = "/path/to/chromedriver"

# Launch the Chrome browser

driver = webdriver.Chrome(driver\_path)

# Navigate to the webpage where the PHP code is located

driver.get("http://localhost/update%201/tech-blog/admin\_panel/add\_category.php")

# Find the category element you want to toggle and click on it

categoryname = "world"

time.sleep(2) # Replace with the name of the category you want to toggle

category\_element = driver.find\_element(By.XPATH, "//td[contains(text(),'world')]/following-sibling::td").click()

time.sleep(2)

# Wait for the page to refresh and verify that the status of the category has been toggled

new\_status = driver.find\_element(By.XPATH, "//td[contains(text(),'world')]/following-sibling::td"). get\_attribute("status")

if new\_status == "1":

print("\n\033[32m\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n Category status has been successfully toggled to active. \n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

else:

print("\n\033[32m\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n Category status has been successfully toggled to inactive. \n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

# Close the browser

driver.quit()

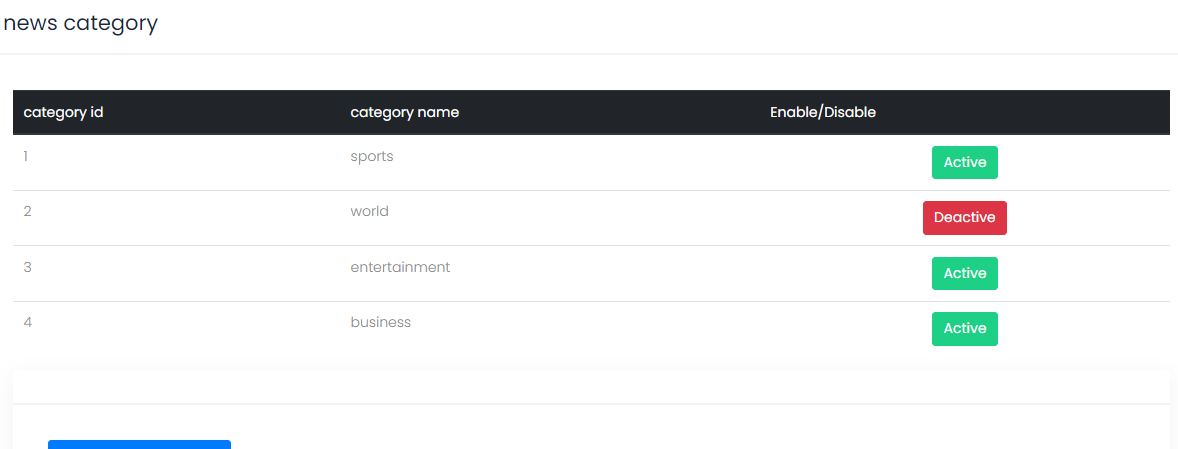


Fig 10: output

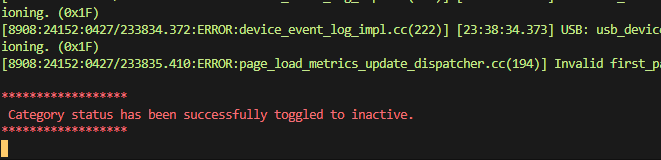
****

Fig 11: Result

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case 2** | | | | | |
| **Project Name: Newzilla** | | | | | |
| **Category Test Case** | | | | | |
| **Test Case ID: Test\_2** | | | **Test Designed By: Rohan** | | |
| **Test Priority(Low/Medium/High):** | | | **Test Designed Date: 24/04/2023** | | |
| **Module Name**: category | | | **Test Executed By: Rohan** | | |
| **Test Title : status of category** | | | **Test Execution Date: 24/04/2023** | | |
| **Description: status of category** | | |  | | |
| **Pre-Condition:** Category should be present | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status(Pass/**  **Fai l)** |
| 1 | Navigate to category page | http://localhost/update%201/tech-blog/admin\_panel/add\_category.php | page loads successfully | page loaded successfully | Pass |
| 2 | Find the category element you want to toggle and click on it | "world" | N/A | N/A | Pass |
| 3 | Wait for the page to refresh and verify that the status of the category... | N/A | If the status of the category has been toggled successfully, print a success message. Otherwise, print a failure message. | N/A | Pass /Fail |
| 4 | Close the browser window | driver.quit() | Browser window is successfully closed | Browser window is successfully closed | Pass |
|  |  |  |  |  |
| **Post-Condition: world is disabled** | | | | | |

**Test Case 3:**

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

# Set the path of the ChromeDriver executable

driver\_path = "/path/to/chromedriver"

# Launch the Chrome browser

driver = webdriver.Chrome(driver\_path)

# Navigate to the webpage where the PHP code is located

driver.get("http://localhost/update%201/tech-blog/admin\_panel/add\_category.php")

# Find the category element you want to toggle and click on it

time.sleep(2) # Replace with the name of the category you want to toggle

add\_button = driver.find\_element(By.ID, "test")

add\_button.click()

time.sleep(2)

# Wait for the page to refresh and verify that the status of the category has been toggled

category\_input = driver.find\_element(By.ID, "categoryname")

category\_input.send\_keys("test")

time.sleep(2)

submit\_button = driver.find\_element(By.ID, "submit")

submit\_button.click()

driver.get("http://localhost/update%201/tech-blog/admin\_panel/add\_category.php")

new\_status = driver.find\_element(By.XPATH, "//td[contains(text(),'test')]").text

if(new\_status=='test'):

print("\n\033[32m\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n Testing success \n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

else:

print("\n\033[31m\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n Testing failed \n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

# Close the browser

driver.quit()

# 

# 

# Fig 12: result

# 

# Fig 13: output

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case 3** | | | | | |
| **Project Name: Newzilla** | | | | | |
| **Category\_add Test Case** | | | | | |
| **Test Case ID: Test\_3** | | | **Test Designed By: Rohan** | | |
| **Test Priority(Low/Medium/High):** | | | **Test Designed Date: 24/04/2023** | | |
| **Module Name**:  **category** | | | **Test Executed By: Rohan** | | |
| **Test Title : adding category add** | | | **Test Execution Date: 24/04/2023** | | |
| **Description: adding a new category** | | |  | | |
| **Pre-Condition:** Category name is not present | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status(Pass/**  **Fai l)** |
| 1 | Navigate to the webpage | http://localhost/update%201/tech-blog/admin\_panel/add\_category.php | page loads successfully | page loaded successfully | Pass |
| 2 | Click on Add Category button | N/A | Add Category form should be displayed | Add Category form is displayed | Pass |
| 3 | Enter category name | "test" | Category name should be entered in the input field | Category name is entered in the input field | Pass |
| 4 | Click on Submit button | N/A | Category should be added to the database | Category is added to the database | Pass |
| 5 | Verify the new category is added | "test" | The new category should be displayed in the category table | N/A | Pass |
| 6 | Close the browser window | driver.quit() | Browser window is successfully closed | Browser window is successfully closed | Pass |
|  |  |  |  |  |
| **Post-Condition: test is created in category** | | | | | |

**Test Case 4:**

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

# Set the path of the ChromeDriver executable

driver\_path = "/path/to/chromedriver"

# Launch the Chrome browser

driver = webdriver.Chrome(driver\_path)

time.sleep(2)

driver.get("http://localhost/miniproject/login1.php")

driver.set\_window\_size(1536,1200)

username\_input = driver.find\_element(By.ID, "name")

username\_input.send\_keys("ROHAN")

time.sleep(2)

password\_input = driver.find\_element(By.ID, "password")

password\_input.send\_keys("a12345678")

time.sleep(3)

submit\_button = driver.find\_element(By.ID, "submit")

submit\_button.click()

time.sleep(1)

driver.get("http://localhost/miniproject/tech-single.php?id=28")

time.sleep(2)

comm\_input = driver.find\_element(By.NAME, "comm")

comm\_input.send\_keys("great")

time.sleep(3)

submit\_button = driver.find\_element(By.NAME, "sub")

driver.execute\_script("arguments[0].click();", submit\_button)

driver.get("http://localhost/miniproject/tech-single.php?id=28")

time.sleep(2)

new\_status = driver.find\_element(By.XPATH, "//p[contains(text(),'great')]").text

if(new\_status=='great'):

print("\n\033[32m\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n Commented added \n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

else:

print("\n\033[31m\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n comment not added \n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

# Close the browser

driver.quit()

# 

# Fig 14: result

# 

# Fig 14: output

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case 4** | | | | | |
| **Project Name: Newzilla** | | | | | |
| **Category\_add Test Case** | | | | | |
| **Test Case ID: Test\_4** | | | **Test Designed By: Rohan** | | |
| **Test Priority(Low/Medium/High):** | | | **Test Designed Date: 26/04/2023** | | |
| **Module Name**:  **comment** | | | **Test Executed By: Rohan** | | |
| **Test Title : adding comment** | | | **Test Execution Date: 26/04/2023** | | |
| **Description: adding user comment** | | |  | | |
| **Pre-Condition:** user should be present | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status(Pass/**  **Fai l)** |
| 1 | Navigate to the webpage | "http://localhost/miniproject/tech-single.php?id=28 | page loads successfully | page loaded successfully | Pass |
| 2 | User navigates to a specific article page | N/A | User should be able to access the article page | User was able to access the article page | Pass |
| 3 | User adds a comment on the article | "good” | comment should be entered in the input field | comment is entered in the input field | Pass |
| 4 | Click on Submit button | N/A | comment is added to the database | comment is added to the database | Pass |
| 5 | Close the browser window | driver.quit() | Browser window is successfully closed | Browser window is successfully closed | Pass |
|  |  |  |  |  |
| **Post-Condition: test is created in category** | | | | | |

# CHAPTER 6

# IMPLEMENTATION

## INTRODUCTION

Implementation is the process of translating software designs and specifications into a functional software system. It involves the actual coding, testing, and integration of software components to create a working software product.

The implementation phase is a critical step in the software development life cycle (SDLC) because it involves turning the abstract ideas and plans into tangible software products that can be used by end-users. The implementation process involves several steps, including:

* Coding: Writing code in the programming language specified in the design phase.
* Testing: Ensuring that the code functions correctly and meets the design specifications. Testing can include unit testing, integration testing, and system testing.
* Debugging: Identifying and fixing any errors or defects in the code.
* Integration: Combining the individual software components into a cohesive system that functions as intended.
* Deployment: Installing the software on the target hardware and making it available to end-users.

During the implementation phase, it is important to follow coding and testing standards to ensure the software is reliable, maintainable, and scalable. It is also important to document the code and maintain version control to facilitate future maintenance and updates. Collaboration between developers and other stakeholders, such as testers, project managers, and end-users, can help to ensure that the software meets the needs and expectations of all stakeholders

## IMPLEMENTATION PROCEDURES

The implementation phase of the software development life cycle (SDLC) involves a series of procedures that must be followed to ensure that the software product is developed according to the design specifications and meets the quality and performance standards. Below are some of the typical procedures involved in the implementation phase:

* Coding: This involves writing the actual code for the software using the programming language specified in the design phase. Developers must follow coding standards and guidelines to ensure that the code is maintainable and scalable.
* Testing: Testing is a critical part of the implementation phase, and it involves testing the software at different levels, including unit testing, integration testing, and system testing. Testers must develop test cases and test scripts based on the design specifications to ensure that the software functions correctly.
* Debugging: During testing, issues and bugs may be identified in the software code. Developers must identify and fix these errors or defects to ensure that the software functions as intended.
* Integration: After individual software components have been tested and debugged, they must be integrated into a cohesive system. The integration process must be carefully managed to ensure that the software functions as intended and that there are no conflicts or errors.
* Deployment: Once the software has been integrated, it must be deployed to the target hardware and made available to end-users. Deployment can be a complex process that involves several steps, including installation, configuration, and testing.
* Documentation: It is essential to document the software code and the implementation procedures to ensure that the software can be easily maintained and updated in the future. Documentation can include user manuals, technical documentation, and code comments.
* Version control: Version control is critical during the implementation phase to ensure that changes to the software code are tracked and managed properly. Version control tools such as Git or SVN can be used to manage code changes and track different versions of the software product.

By following these procedures, software developers can ensure that the software product is developed according to the design specifications and meets the quality and performance standards required by the end-users.

### User Training

User training is a critical component of any software development project. It involves providing end-users with the necessary knowledge and skills to effectively use the software product. The goal of user training is to ensure that end-users can use the software product efficiently and effectively to meet their business needs.

Below are some common steps involved in user training:

• Identify training needs: Before training can begin, it is essential to identify the training needs of the end-users. This can be done by conducting a needs analysis, which involves assessing the current skills and knowledge of the end-users and identifying any gaps that need to be addressed.

• Develop training materials: Once the training needs have been identified, training materials can be developed. This can include user manuals, online tutorials, and training videos. The training materials should be designed to be easy to understand and follow, and should cover all the necessary features and functionalities of the software product.

• Conduct training sessions: Training sessions can be conducted in-person or online, depending on the needs of the end-users. The training sessions should be interactive and engaging, and should provide opportunities for end-users to ask questions and practice using the software product.

• Assess learning: After the training sessions, it is important to assess whether the end-users have learned the necessary skills and knowledge to effectively use the software product. This can be done through quizzes, assessments, or practical exercises.

• Provide ongoing support: Even after the initial training, it is important to provide ongoing support to end-users. This can include a helpdesk or support team that can provide assistance when needed, as well as updates to the training materials to reflect any changes or updates to the software product.

Effective user training can help to ensure that end-users are able to use the software product efficiently and effectively, which can lead to increased productivity and improved business outcomes.

### Training on the Application Software

Training on application software is an essential component of software implementation. It is the process of providing end-users with the knowledge and skills necessary to operate and use the application software effectively. The goal of training is to ensure that end-users are able to utilize all the features and functionalities of the software to perform their tasks and achieve their goals.

Below are some best practices for providing effective training on application software:

• Identify the target audience: Before developing training materials, it is essential to identify the target audience and their specific needs. Different groups of end-users may require different levels of training or different types of training materials.

• Develop training materials: Training materials can include user manuals, online tutorials, videos, and live training sessions. The training materials should be designed to be easy to understand and follow, and should cover all the necessary features and functionalities of the software product.

• Use interactive training methods: Interactive training methods such as demonstrations, simulations, and hands-on exercises can help end-users to better understand and retain the training materials.

• Provide ongoing support: After the initial training, it is important to provide ongoing support to end-users. This can include a helpdesk or support team that can provide assistance when needed, as well as updates to the training materials to reflect any changes or updates to the software product.

• Evaluate training effectiveness: It is important to evaluate the effectiveness of the training by assessing the end-users' knowledge and skills before and after the training. This can help identify areas that may require additional training or support.

• Provide refresher training: As the software product evolves, it is important to provide refresher training to end-users to ensure that they are up-to-date with the latest features and functionalities.

Effective training on application software can help to ensure that end-users are able to use the software product effectively, which can lead to increased productivity and improved business outcomes.

### System Maintenance

System maintenance is the process of keeping a software system up-to-date and functioning properly after it has been deployed. It involves a variety of activities that are designed to ensure that the system remains stable, secure, and efficient over time. Some of the key activities involved in system maintenance include:

• Updates and patches: One of the most important aspects of system maintenance is keeping the system up-to-date with the latest software updates and security patches. These updates and patches are released periodically by software vendors to fix bugs, add new features, and address security vulnerabilities. Applying these updates in a timely manner is essential to ensure that the system remains secure and reliable.

• Performance monitoring: Regular performance monitoring is another key aspect of system maintenance. This involves tracking system performance metrics such as CPU usage, memory utilization, and network traffic to identify potential issues and optimize system performance.

• Backup and recovery: Backing up system data and ensuring that it can be recovered in the event of a system failure or data loss is an important part of system maintenance. This involves creating regular backups of system data and testing the recovery process to ensure that it works as expected.

• Security management: Maintaining system security is critical to protecting sensitive data and preventing unauthorized access to the system. This involves implementing security controls such as firewalls, antivirus software, and access controls, and regularly monitoring the system for potential security breaches.

### Hosting

Hosting refers to the process of storing and serving website files and data on a server,  
which is accessible to the internet. Website hosting is necessary for businesses or  
individuals who want to make their website accessible to the public. Hosting providers also offer various tools and features, including website builders, one-click installations, and security measures. A reliable hosting service is crucial for ensuring website accessibility, speed, and security.

• Create an account in the online free webhosting site https://in.000webhost.com/

• After creating the account by selecting the domain name the main page will show the

number of active account. Current active account is the account that we created this

time.

• By clicking the link it will show the home page.

• Once you have created an account, you can use the free website builder tool to create and design your website or upload your existing website files to the platform.

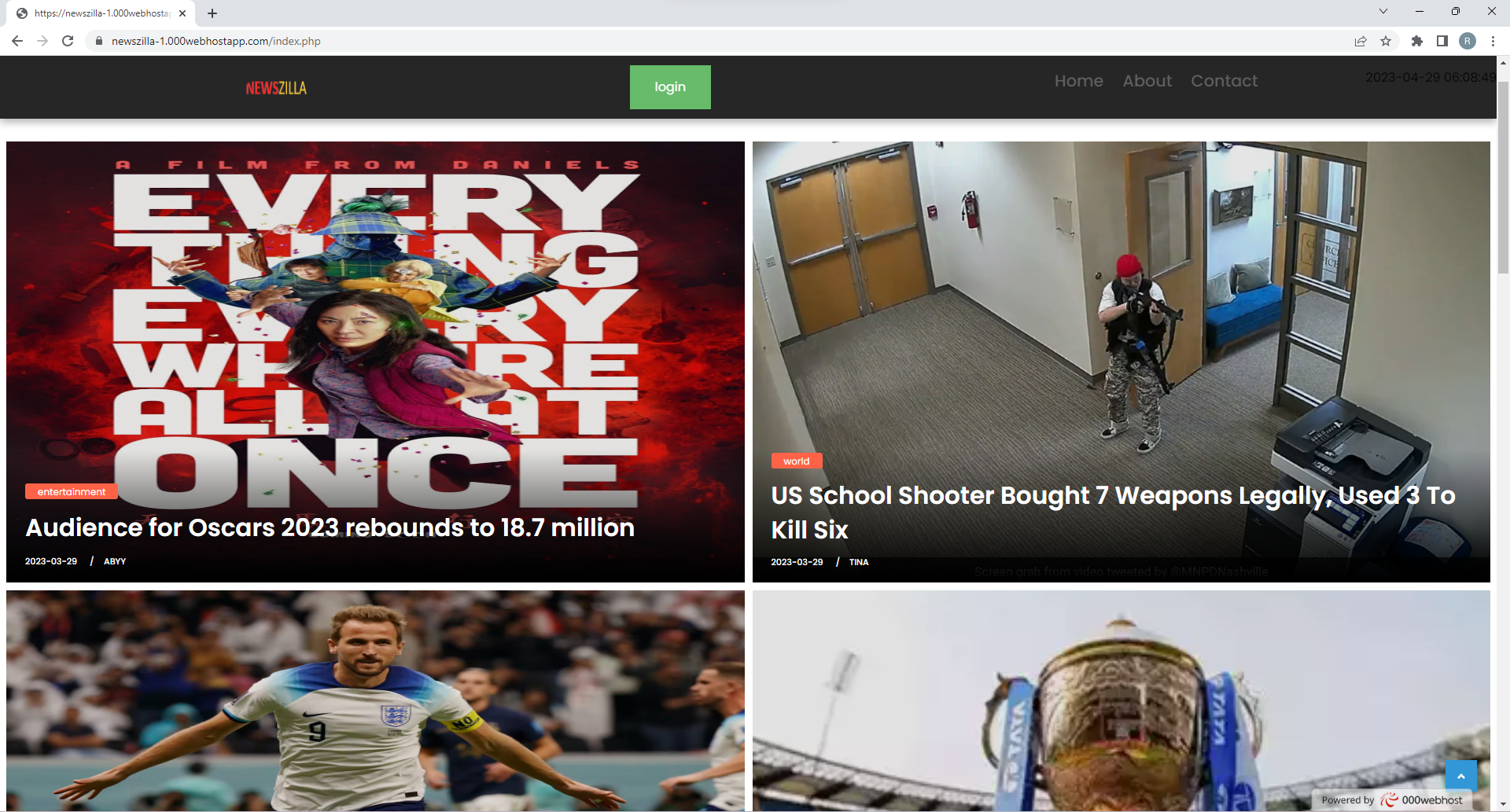
• Then upload the files in htdocs in our local system to the setup.

• Click the MySQL databases link in the i then create a database. Then click on the phpMyAdmin and import the database file to it. Confirm that all the tables are imported.

• Open the File Manager in the site and edit the connection page by renaming the database

• In the home page of the site the link will be visible and then we can copy and paste in

the URL on the search engine for accessing the website



# CHAPTER 7

# CONCLUSION AND FUTURE SCOPE

## CONCLUSION

## 

. In conclusion, an online news portal is an essential software tool that can greatly simplify the publishing of a news. This system can help to streamline a range of business processes, from news management and sales tracking to reader management and. By automating these processes and providing real-time data and analytics, this system can help to increase efficiency, reduce costs, and improve customer satisfaction.

The development and implementation of, an online news portal System involve a range of activities, from requirements gathering and system design to testing and deployment. It is important to ensure that the system is properly designed, tested, and maintained to ensure that it remains reliable, secure, and efficient over time.

Overall, a well-designed, an online news portal can provide significant benefits to reporter who like to publish the news to wide range of people. By improving business processes and optimizing operations, this system can help reporters to increase their interaction, improve customer satisfaction, and achieve long-term success in a competitive marketplace.

* 1. **FUTURE SCOPE**

The future of online news portals looks very promising as the world becomes increasingly connected through the internet. Here are some potential areas of growth and development:

Personalization: With advancements in AI and machine learning, online news portals will become more personalized to cater to individual user preferences. Users will be able to see news articles that are relevant to their interests, based on their past behavior and reading history.

Multimedia Content: Online news portals will continue to focus on multimedia content such as videos, podcasts, and infographics, as they are more engaging and interactive for users.

Social Media Integration: Online news portals will continue to integrate with social media platforms to reach a wider audience and allow users to share news articles with their friends and followers.

Mobile Optimization: As more people access the internet through mobile devices, online news portals will need to optimize their content for mobile viewing, ensuring that their websites and apps are mobile-friendly.

Subscription and Paywalls: Online news portals will continue to explore subscription-based models and paywalls as a way to generate revenue and sustain their businesses.

.

**CHAPTER 8**

**BIBLIOGRAPHY**

REFERENCES:

* + Gary B. Shelly, Harry J. Rosenblatt, “System Analysis and Design”, 2009.
  + Roger S Pressman, “Software Engineering”, 1994.
  + PankajJalote, “Software engineering: a precise approach”, 2006.
  + James lee and Brent ware Addison, “Open source web development with LAMP”,2003
  + The Complete reference PHP by Steven Holzner
  + The Complete reference MySQL by Vikram Vaswani
  + CSS Cookbook byChristopher Schmitt

### WEBSITES:

* + - <https://getbootstrap.com/>
    - <https://fontawesome.com/>
    - <https://www.w3schools.com/>
    - <https://getcssscan.com/>

# CHAPTER 9

# APPENDIX

## Sample Code

**News view, User complaint, comment**

<!DOCTYPE html>

<html lang="en">

<?php

require\_once 'navbar1.php';

$news\_id = $\_SESSION['news\_id'];

$news\_id= ($\_GET["id"]);

$user= $\_SESSION['username'];

?>

<!-- Basic -->

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<!-- Mobile Metas -->

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Design fonts -->

<link href="https://fonts.googleapis.com/css?family=Poppins:300,400,500,600,700" rel="stylesheet">

<!-- Bootstrap core CSS -->

<link href="css/bootstrap.css" rel="stylesheet">

<!-- FontAwesome Icons core CSS -->

<link href="css/font-awesome.min.css" rel="stylesheet">

<link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet" >

<!-- Custom styles for this template -->

<link href="style.css" rel="stylesheet">

<!-- Responsive styles for this template -->

<link href="css/responsive.css" rel="stylesheet">

<!-- Colors for this template -->

<link href="css/colors.css" rel="stylesheet">

<!-- Version Tech CSS for this template -->

<link href="css/version/tech.css" rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" s></script>

<style>

html {

scroll-behavior: smooth;

}

.modal-body{

padding:120px;

z-index: 2;

}

.close{

color:red;

}

.dropdwn{

padding:12px;

}

.save{

border-radius: 5px;

background: blue;

padding: 12px;

color: white;

}

.g-height-50 {

height: 50px;

}

.g-width-50 {

width: 50px !important;

}

@media (min-width: 0){

.g-pa-30 {

padding: 2.14286rem !important;

}

}

.g-bg-secondary {

background-color: rgba(225, 194, 169, 0.35) !important;

}

.u-shadow-v18 {

box-shadow: 0 5px 10px -6px rgba(0, 0, 0, 0.99);

}

.g-color-gray-dark-v4 {

color: #777 !important;

}

.g-font-size-12 {

font-size: 0.85714rem !important;

}

.media-comment {

margin-top:20px

}

.img-fluidn{

height: 800px;

}

</style>

</head>

<body>

<div id="wrapper">

<?php

$con = mysqli\_connect("localhost", "root", "", "mindb");

// Check connection

if (mysqli\_connect\_errno())

{

echo "Failed to connect to MySQL: " . mysqli\_connect\_error();

}

$q="UPDATE NEWS SET view\_count= view\_count+1 WHERE news\_id ='$news\_id' ";

$r= mysqli\_query($con, $q);

$query1 = "SELECT news\_id, category\_id, headline, news\_con, img, reporter\_id,DATE(date) AS date1, view\_count,status FROM news where news\_id ='$news\_id' ";

$result= mysqli\_query($con, $query1);

while($row = mysqli\_fetch\_array($result))

{

$query2 = "SELECT \* FROM reporterdetails where reporter\_id =$row[reporter\_id] ";

$result2= mysqli\_query($con, $query2);

$row2 = mysqli\_fetch\_array($result2) ;

$query3= "SELECT categoryname ,id FROM category where id =$row[category\_id] ";

$result3= mysqli\_query($con, $query3);

$row3 = mysqli\_fetch\_array($result3) ;

$head=$row['headline'];

$cont=$row['news\_con'];

$date=$row['date1'];

$img=$row['img'];

$reporter=$row2['user'] ;

$reporter\_id=$row2['reporter\_id'] ;

$count=$row['view\_count'];

$complete\_post=$row['news\_con'];

$id=$row3['id'];

$pic=$row2['profilepic'];

$count1=$row['view\_count']+1;

$query4= "UPDATE NEWS set view\_count='$' where id =$row[category\_id] ";

$result3= mysqli\_query($con, $query3);

?>

<section class="section single-wrapper">

<div class="container">

<div class="row">

<div class="col-lg-9 col-md-12 col-sm-12 col-xs-12">

<div class="page-wrapper">

<div class="blog-title-area text-center">

<span class="color-orange"><a href="tech-category.php?categoryid=<?php echo $id; ?> "><?php echo "$row3[categoryname]";?></a></span>

<h3><?php echo"$head" ?></h3>

<div class="blog-meta big-meta">

<small><a href="tech-single.html" title=""><?php echo"$date" ?></a></small>

<small><a href="tech-author.html" title="">by <?php echo"$reporter" ?></a></small>

<small><a href="#" title=""><i class="fa fa-eye"></i> <?php echo"$count" ?></a></small>

</div><!-- end meta -->

<div class="post-sharing">

<ul class="list-inline">

<li><a href="#" class="fb-button btn btn-primary"><i class="fa fa-facebook"></i> <span class="down-mobile">Share on Facebook</span></a></li>

<li><a href="#" class="tw-button btn btn-primary"><i class="fa fa-twitter"></i> <span class="down-mobile">Tweet on Twitter</span></a></li>

<li><a href="#" class="gp-button btn btn-primary"><i class="fa fa-google-plus"></i></a></li>

</ul>

</div><!-- end post-sharing -->

</div><!-- end title -->

<P>DO YOU FIND THE CONTENT INAPPROPRIATE ? <a hre>

<!-- Button trigger modal -->

<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">

Send your concern

</button>

<!-- Modal -->

<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel"></h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<form class="form-wrapper" action="#" method="POST">

Select your reason

<select name="reason" class="dropdwn">

<option value="Violent/Replusive Content" >Violent/Replusive Content</option>

<option value="Hateful Content" >Hateful Content</option>

<option value="Misinformation" >Misinformation</option>

<option value="Promote Terrorism" >Promote Terrorism</option>

<option value="Spam/Misleading" >Spam/Misleading</option>

</select>

</div>

<div class="modal-footer">

<button type="submit" class="save" name="ban">Save changes</button>

</form>

<?php if (isset($\_POST['ban'])) {

$reason=$\_POST['reason'];

$queryr = "INSERT INTO news\_ban(news\_id,reason,user\_id) VALUES ('$news\_id','$reason','$user\_id')";

mysqli\_query($con, $queryr);

?>

<script>

location.replace("tech-single.php?id=<?php echo $news\_id; ?>")

</script>

<?php

}

?>

</div>

</div>

</div>

</div>

<div class="single-post-media">

<img src="./reporter/upload/thumbnail/<?php echo "$row[img]"; ?>" alt="image" class="img-fluidn">

</div><!-- end media -->

<div class="blog-content">

<div class="pp">

<p> <?php echo" $complete\_post" ?></p>

</div><!-- end pp -->

</div><!-- end content -->

<?php }?>

<div class="blog-title-area">

<div class="tag-cloud-single">

<span>Tags</span>

<?php

$query1 = "select categoryname,id,status from category";

$result= mysqli\_query($con, $query1);

while($row = mysqli\_fetch\_array($result))

{

if($row['status']=='1'){

$id=$row['id'];

?>

<small><a href="tech-category.php?categoryid=<?php echo $id; ?> "><?php echo "$row[categoryname]";?></a></small>

<?php }

}?>

</div><!-- end meta -->

</div><!-- end title -->

<hr class="invis1">

<hr class="invis1">

<div class="custombox authorbox clearfix">

<h4 class="small-title">About author</h4>

<div class="row">

<div class="col-lg-2 col-md-2 col-sm-2 col-xs-12">

<img src="./reporter/uploads/<?php echo "$pic"; ?>" alt="img" class="img-fluid rounded-circle">

</div><!-- end col -->

<?php

$query1 = "select headline from news where reporter\_id='$reporter\_id'";

$result= mysqli\_query($con, $query1);

$row = mysqli\_num\_rows($result)

?>

<div class="col-lg-10 col-md-10 col-sm-10 col-xs-12">

<h4><a href="#"></a><?php echo"$reporter" ?></h4>

<p>qualification : <?php echo $row2['qualification']; ?></p>

<p>No of news published : <?php echo $row; ?></p>

</div><!-- end col -->

</div><!-- end row -->

</div><!-- end author-box -->

<hr class="invis1">

<hr class="invis1">

<div class="custombox clearfix">

<div class="custombox clearfix">

<h4 class="small-title">Leave a comment</h4>

<div class="row">

<div class="col-lg-12">

<form class="form-wrapper" action="#" method="POST">

<textarea class="form-control" placeholder="Your comment" class="com" name="comm"></textarea>

<button type="submit" class="btn btn-primary" name="sub">Submit Comment</button>

</form>

</div>

</div>

</div>

<?php

if (isset($\_POST['sub'])) {

$comm=$\_POST['comm'];

$query1 = "INSERT INTO comment(user\_id,news\_id,comment) VALUES ('$user\_id','$news\_id','$comm')";

mysqli\_query($con, $query1);

?>

<script>

location.replace("tech-single.php?id=<?php echo $news\_id; ?>")

</script>

<?php

}

?>

<h4 class="small-title">Comments</h4>

<div class="row">

<div class="col-lg-12">

<div class="comments-list">

<?php

$query1 = "SELECT user\_id,comment FROM comment WHERE news\_id ='$news\_id' ";

$result= mysqli\_query($con, $query1);

while($row = mysqli\_fetch\_array($result))

{

$user\_id = $row['user\_id'];

$com= $row['comment'];

$query2 = "SELECT \* FROM userdetails WHERE userid='$user\_id'";

$result2 = mysqli\_query($con, $query2);

$row2 = mysqli\_fetch\_assoc($result2);

$img = $row2['photo'];

$name = $row2['user'];

?>

div class="container">

<div class="row">

<div class="col-md-8">

<div class="media g-mb-30 media-comment">

<img class="d-flex g-width-50 g-height-50 rounded-circle g-mt-3 g-mr-15" src="../uploads/<?php echo $img; ?>" alt="Image Description">

<div class="media-body u-shadow-v18 g-bg-secondary g-pa-30">

<div class="g-mb-15">

<h5 class="h5 g-color-gray-dark-v1 mb-0"><?php echo $name ?></h5>

</div>

<br><br>

<p><?php echo $com ?></p>

</div>

</div>

</div>

<?php } ?>

</div>

</div><!-- end col -->

</div><!-- end row -->

</div><!-- end custom-box -->

<hr class="invis1">

</div><!-- end page-wrapper -->

</div><!-- end col -->

<?php mysqli\_close($con) ?>

</div>

</div>

<div class="banner">

<span class="close" onclick="this.parentElement.style.display='none';">&times;</span>

<img src="ad.jpg" alt="Banner image">

</div><!-- end row -->

</div><!-- end container -->

</section>

<footer class="footer">

<div class="container">

<div class="row">

<div class="col-lg-7">

<div class="widget">

<div class="footer-text text-left">

<a href="index.html"><img src="logo1.png" alt="" class="img-fluid"></a>

<p>NEWSZILLA is a online news portal </p>

<div class="social">

<a href="#" data-toggle="tooltip" data-placement="bottom" title="Facebook"><i class="fa fa-facebook"></i></a>

<a href="#" data-toggle="tooltip" data-placement="bottom" title="Twitter"><i class="fa fa-twitter"></i></a>

<a href="#" data-toggle="tooltip" data-placement="bottom" title="Instagram"><i class="fa fa-instagram"></i></a>

<a href="#" data-toggle="tooltip" data-placement="bottom" title="Google Plus"><i class="fa fa-google-plus"></i></a>

<a href="#" data-toggle="tooltip" data-placement="bottom" title="Pinterest"><i class="fa fa-pinterest"></i></a>

</div>

<hr class="invis">

</div><!-- end footer-text -->

</div><!-- end widget -->

</div><!-- end col -->

<div class="col-lg-3 col-md-12 col-sm-12 col-xs-12">

</div><!-- end col -->

<div class="col-lg-2 col-md-12 col-sm-12 col-xs-12">

<div class="widget">

<div class="link-widget">

<ul>

<li><a href="#">About us</a></li>

<li><a href="#">Advertising</a></li>

</ul>

</div><!-- end link-widget -->

</div><!-- end widget -->

</div><!-- end col -->

</div>

<div class="dmtop">Scroll to Top</div>

</div><!-- end container -->

</footer><!-- end footer -->

</div><!-- end wrapper -->

</body>

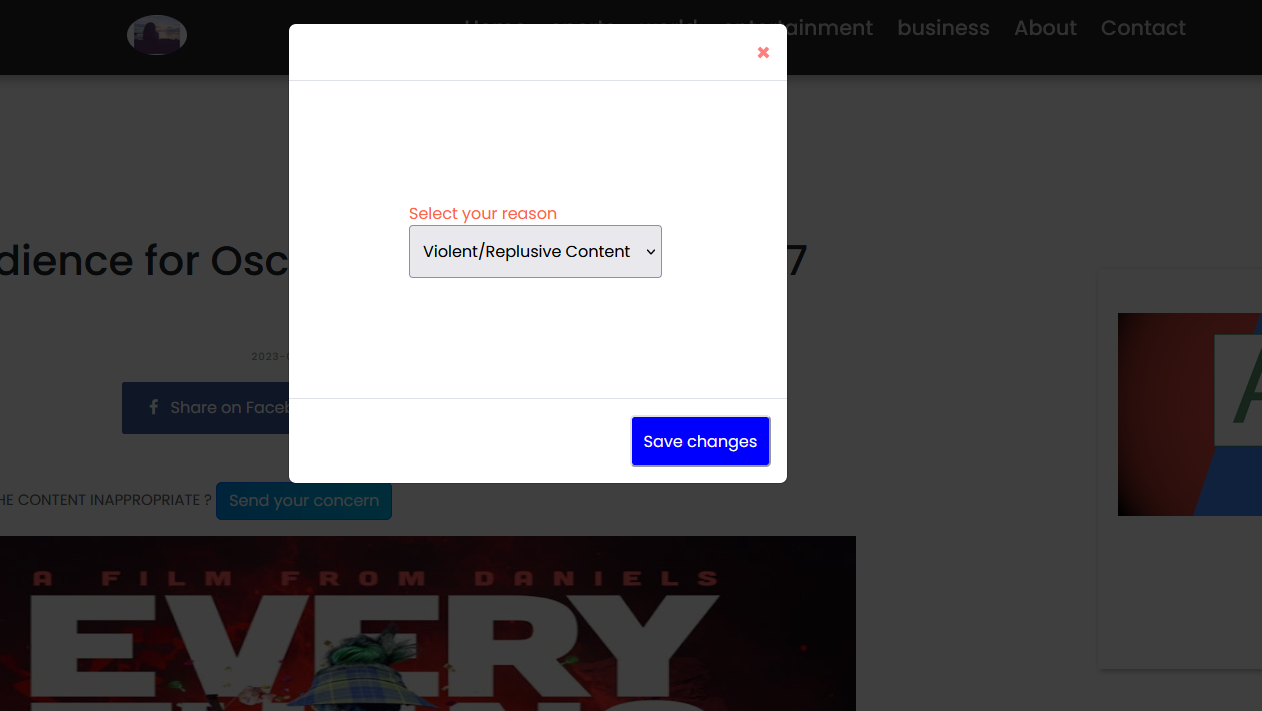
</html>

## Screen Shots

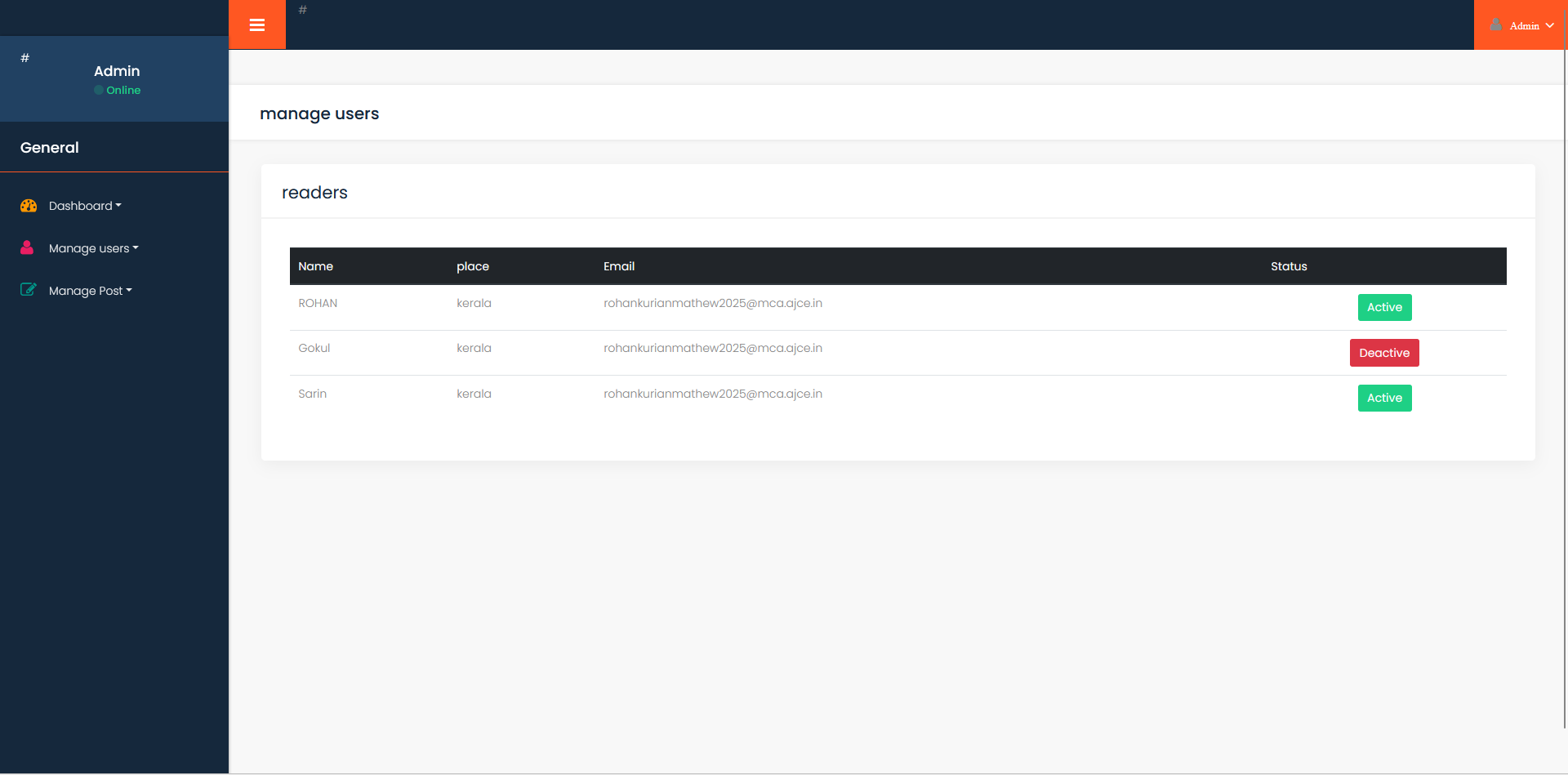
## login

## 

## User complaint

****

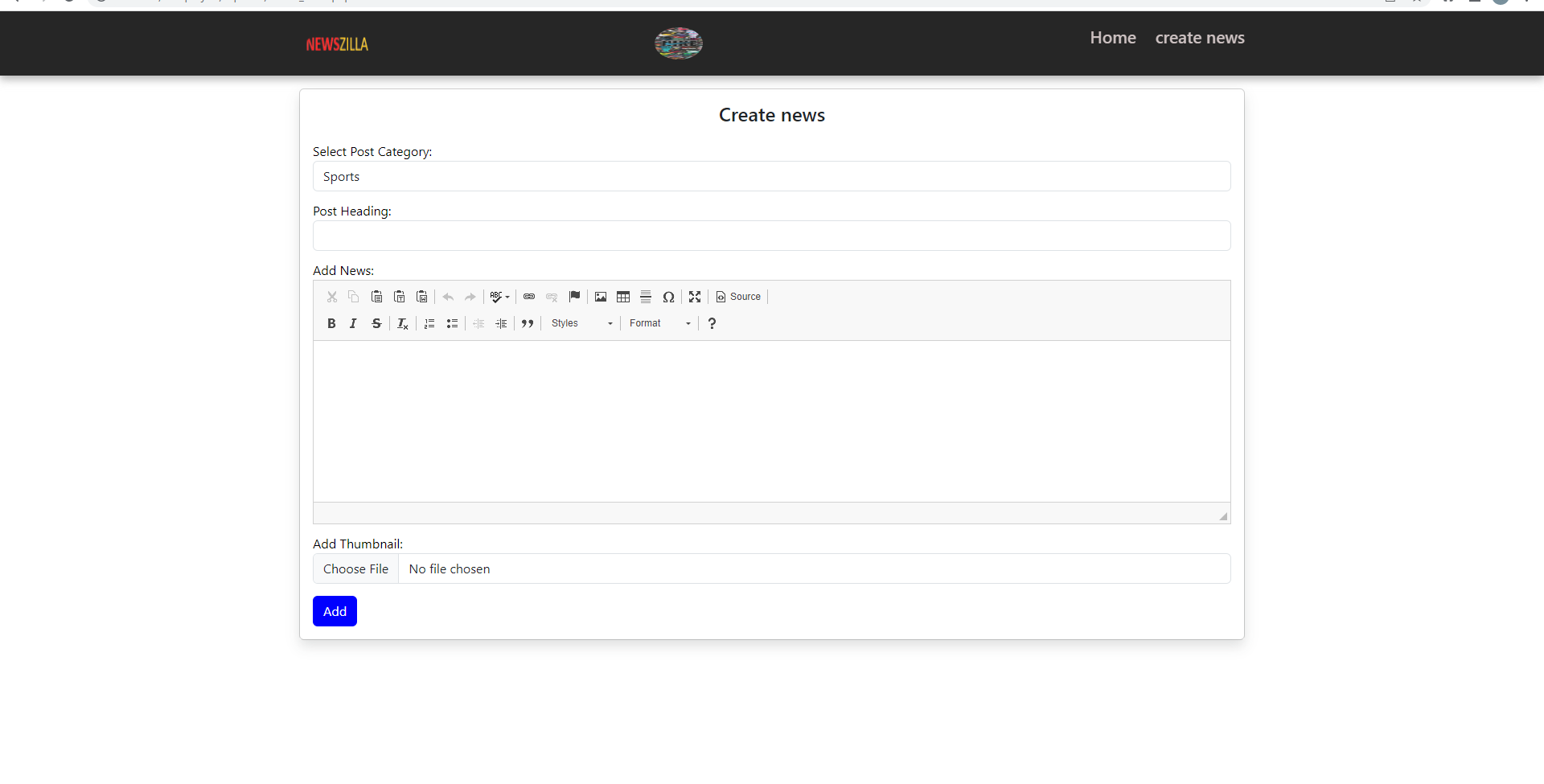
**Admin user manage**



**Reporter details and comment section for reader**



**Creating news**



**Managing news**

