A

Major Project Report on

SMART BUS PASS MANAGEMENT SYSTEM

Submitted for partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

Submitted by

M.VAISHNAVI	**21K81A12A0**
A.AADHYA	**21K81A1266**
R.ARJUN	**21K81A12B2**

Under the Guidance of

Mrs. M. SANDHYA RANI

ASSISTANT PROFESSOR

DEPARTMENT OF INFORMATION TECHNOLOGY

St. MARTIN'S ENGINEERING COLLEGE

UGC Autonomous
Affiliated to JNTUH, Approved by AICTE,
Accredited by NBA & NAACA+, ISO 9001- 2008 Certified Dhulapally,
Secunderabad- 500100
www.smec.ac.in

MARCH–2025

St. MARTIN'S ENGINEERING COLLEGE
UGCAutonomous
Affiliated to JNTUH, Approved by AICTE NBA &
NAAC A+ Accredited Dhulapally, Secunderabad
- 500 100 www.smec.ac.in

Certificate

This is to certify that the project entitled **“SMART BUS PASS MANAGEMENT SYSTEM”** is being submitted by M.Vaishnavi (21K81A12A0), A.Aadhya (21K81A1266), R.Arjun (21K81A12B2) in fulfilment of the requirement for the award of degree of **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** is recorded of Bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Signature of Guide
Mrs. M. SANDHYARANI
Assistant Professor
Department of Information Technology

Signature of HOD
Dr. N. KRISHNAIAH
Professor and Head of Department
Department of InformationTechnology

Internal Examiner

External Examiner

Place: Date:

St. MARTIN'S ENGINEERING COLLEGE
UGC Autonomous
Affiliated to JNTUH, Approved by AICTE NBA &
NAAC A+ Accredited Dhulapally, Secunderabad
-500 100 www.smec.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

Declaration

We, the students of '**Bachelor of Technology in Department of Information Technology'**,session: 2021 - 2025**, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad,** hereby declare that the work presented in this Project Work entitled **"SMART BUS PASS MANAGEMENT SYSTEM"** is the outcome of our own Bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

M.VAISHNAVI	**21K81A12A0**
A.AADHYA	**21K81A1266**
R.ARJUN	**21K81A12B2**

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded our efforts with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to our College Management for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to **Dr. P. SANTOSH KUMAR PATRA**, Professor and Group Director, St. Martin’s Engineering College Dhulapally, for permitting us to undertake this project.

We wish to record our profound gratitude to **Dr. M. SREENIVAS RAO**, Principal, St. Martin’s Engineering College, for his motivation and encouragement.

We are also thankful to **Dr. N. KRISHNAIAH**, Professor and Head of the Department, Information Technology, St. Martin’s Engineering College, Dhulapally, Secunderabad. for his support and guidance throughout our project as well as Project Coordinator **Mr. T.SURESH**, Assistant Professor, Information Technology department for his valuable support.

We would like to express our sincere gratitude and indebtedness to our project supervisor **Mrs.M.SANDHYARANI**, Assistant Professor, Information Technology, St. Martins Engineering College, Dhulapally, for her support and guidance throughout our project.

Finally, we express thanks to all those who have helped us successfully completing this project. Furthermore, we would like to thank our family and friends for their moral support and encouragement. We express thanks to all those who have helped us in successfully completing the project.

M.VAISHNAVI	**21K81A12A0**
A.AADHYA	**21K81A1266**
R.ARJUN	**21K81A12B2**

ABSTRACT

This paper presents a software application aimed at facilitating the maintenance and renewal of bus passes for passengers who rely on bus services for their daily commutes. The application streamlines the process of obtaining and renewing travel passes, reducing the hassle of purchasing tickets daily. Upon registration and login, users are provided with various options, including acquiring a new pass or renewing an existing one. By selecting the desired route and pass duration, users are presented with the corresponding payment amount. Once the payment is completed, the updated pass is made available under the “Current Pass” section.

This innovative web-based framework enables easy access to transportation pass information online, eliminating the need to physically visit bus stops. Notably, students and travelers benefit from the convenience of obtaining transportation passes through this platform, bypassing the need to wait in queues or purchase individual tickets for each journey. The system supports online credit and professional card payments, enhancing the ease of installment transactions. The application encompasses elements of network servers, information systems, testing, cellular networks, web servers, computer architecture, bus operators, and public transport services, collectively contributing to an efficient and user-friendly transportation pass solution.

LIST OF FIGURES

LIST OF ACRONYMS AND DEFINITIONS

S.NO	ACRONYM	DEFINITION
01.	RFC	Random Forest Classifier
02.	ETC	Extra Tree Classifier
03.	RFA	Random Forest Algorithm
04.	UML	Unified Modelling Language
05.	SMOTE	Synthetic Minority Over-sampling Technique
06.	WORA	Write Once Run Anywhere
07.	JDBC	Java Data Base Connectivity
08.	ODBC	Open Data Base Connectivity
09.	GUI	Graphical User Interface

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Overview:

A **Smart Bus Pass Management System** is an advanced digital solution designed to streamline the issuance, renewal, and verification of bus passes, making public transportation more efficient and user-friendly. This system enables passengers to apply for and renew their bus passes online through a web portal or mobile application, eliminating the need for long queues and paperwork. It incorporates technologies such as QR codes, NFC, or RFID-based smart cards for easy validation by conductors using mobile apps or scanners. Additionally, it supports multiple payment methods, including UPI, credit/debit cards, and digital wallets, ensuring a seamless transaction experience

1.2 Problem Statement:

The traditional bus pass management system faces several challenges, including manual paperwork, long queues, inefficiencies in verification, and risks of fraud or misuse. Passengers often experience difficulties in applying for, renewing, and validating their bus passes, leading to delays and inconvenience. Additionally, the reliance on physical documents increases the chances of loss or damage, making the process cumbersome for both users and transport authorities. Conductors and bus operators struggle with time-consuming manual pass verification, which slows down boarding and affects overall efficiency

1.3 Existing System:

The existing bus pass management system primarily relies on manual or semi-digital processes, which often lead to inefficiencies and inconvenience for both passengers and transit authorities. In most cases, passengers are required to visit physical ticket counters to apply for or renew their bus passes, which involves filling out paperwork, submitting identification documents, and making payments in cash. This traditional method is time-consuming, prone to human errors, and results in long queues, especially during peak renewal periods. Additionally, the verification of bus passes is typically done manually by conductors, increasing the chances of fraud, misuse, or unauthorized travel.

While some transport systems have introduced digital pass generation and online application portals, many still lack automation in validation, tracking, and renewal notifications. In cases where smart cards or barcoded passes are used, the dependency on physical cards means passengers still face issues related to loss, damage, or the need for frequent replacements

1.3 Proposed System:

The proposed **Smart Bus Pass Management System** is a fully digital and automated solution designed to overcome the inefficiencies of the existing system. It enables passengers to apply for, renew, and manage their bus passes through a web portal or mobile application, eliminating the need for physical visits and paperwork. The system generates e-passes with QR codes or smart cards embedded with NFC/RFID technology, allowing quick and secure validation by bus conductors using handheld devices or scanners. Additionally, it supports multiple online payment methods, such as UPI, credit/debit cards, and digital wallets, ensuring a seamless and cashless transaction process..

Applications of Proposed System

- **Public Transportation Management**

Enhances the efficiency of city and state-run bus services by automating pass issuance, renewal.

- **Educational Institutions**

Enables students to avail of subsidized bus passes conveniently, reducing the burden of manualverification by schools and colleges.

- **Corporate & Employee Transport Services**

Facilitates seamless pass management for companies providing transportation to employees, ensuring easy tracking and verification.

.

1.4 Need:

The need for a Smart Bus Pass Management System arises from the inefficiencies and challenges associated with traditional bus pass issuance and validation methods. In the existing system, passengers often face long queues, tedious paperwork, and delays in obtaining or renewing bus passes, leading to inconvenience and time wastage. Manual verification by conductors also slows down the boarding process and increases the risk of fraud or misuse of bus passes. Additionally, the lack of digital integration makes it difficult for transport authorities to track pass usage, optimize routes, and implement subsidy programs effectivel.yA smart system addresses these issues by offering a fully automated, digital, and cashle**ss** solution that simplifies the process for both passengers and transport administrators. With online pass application , QR code or NFC-based validation, real-time bus tracking, and automated notifications, the system ensures greater efficiency, security, and transparency. It also promotes an **eco-friendly approach** by reducing paper-based processes and encourages the adoption of **smart city initiatives** by integrating with IoT-based solutions. Overall, a **Smart Bus Pass Management System** enhances passenger convenience, reduces operational costs, prevents fraud, and contributes to a more sustainable and efficient public transportation system

CHAPTER – 2

LITERATURE SURVEY

[1] In 2020, Akca et al. developed and validated a real-time risk assessment tool for 30-day Stress readmissions, integrated Information Exchange. This tool aimed to provide timely risk assessments by utilizing real-time patient data. Their study demonstrated successful deployment and validation, showing that such tools can enhance the management of patient readmission risks and improve care outcomes. The approach highlights the importance of real-time data integration in healthcare systems to pre-emptively address readmission risks.

[2] In 2021, Ganesh et al. Walsh and colleagues explored various calibration methods for predictive models of Stress readmission risk, focusing on the models' clinical usefulness and discrimination capabilities. They compared different methodologies to assess how well these models predict readmission risks and their practical application in clinical settings. The study emphasizes that beyond accurate risk discrimination, predictive models must also be clinically relevant and actionable to improve crop care and management.

[3] Golas et al. (2018) Golas et al. applied machine learning techniques to predict 30-day readmissions for stress crops using electronic records. Their retrospective analysis showcased the potential of machine learning models to identify stress conditions accurately. By leveraging extensive medical data, the study underscores the value of data-driven approaches in predicting readmissions and highlights the promise of machine learning in chronic disease management.

[4] Reddy & Delen (2018) Reddy and Delen utilized RNN-LSTM deep learning methodologies to predict stress conditions for crops.Their study demonstrated how recurrent neural networks (RNNs) and long short-term memory (LSTM) models could analyse sequential patient data to forecast readmissions effectively. This work illustrates advancements in deep learning techniques and their application to chronic disease prediction.

[5] Huang et al. (2020) Huang used a transformer-based model designed to analyse stress condition of crops. Byincorporating natural language processing (NLP) with deep learning.This study highlights the potential of combining NLP techniques with predictive modelling to enhance the accuracy and effectiveness of readmission predictions.

[6] In 2022, Timothy Malche et.al. used recurrent neural network models to detect early signs of heart failure onset. Their study demonstrated the effectiveness of RNNs in analysing temporal patient data to identify early indicators of crop. This approach emphasizes the capability of advanced machine learning models to provide early warnings and improve crop management through timely interventions.

[7] Choi et al. (2016) In their "Doctor AI" study, Choi et al. employed RNNs to predict various crop events based on stress conditions. The research focused on using deep learning to enhance crop event predictions, showcasing the potential of RNNs in improving predictive accuracy. The study reflects the growing application of machine learning in crop conditions.

[8] Choi et al. (2016) Choi et al. explored stress concept representation. Their work demonstrated how learning from unstructured data could improve predictive modeling. By representing different concepts effectively, their approach enhanced prediction accuracy and underscored the value of advanced data representation techniques in analyzing crop stress.

[9] In 2020, Nguyen et al. developed Deepr, a convolutional neural network (CNN) model for analysing records. Their study highlighted the ability of CNNs to extract and interpret features from complex datasets. Deepr's success in record analysis reflects the effectiveness of CNNs in enhancing predictive modelling and improving performance and better data handling.

[10]In 2020, Quan et al. focused on coding algorithms .Their work provided essential methodologies for accurately identifying stress conditions, which are crucial for risk adjustment and predictive modelling.

CHAPTER 3

EXISTING SYSTEM

3.1 Traditional Methods:

The **traditional methods** for bus pass management rely heavily on **manual and paper-based** processes, which are often inefficient and time-consuming. In most cases, passengers need to visit designated counters, fill out application forms, submit physical identification documents, and make payments in cash to obtain or renew their bus passes. This process results in long queues, delays, and inconvenience, especially during peak hours. Once issued, these passes are typically in the form of **paper-based cards or laminated tickets**, which are prone to damage, loss, or forgery.

Verification of passes is usually done manually by bus conductors, who visually inspect them, increasing the risk of **human error and fraud**, as counterfeit or expired passes can be used without detection. Additionally, there is **no centralized database** to track pass usage, making it difficult for transport authorities to analyze travel patterns, optimize routes, or implement subsidy programs efficiently. While some systems have introduced **semi-digital solutions** such as printed barcodes or magnetic stripe cards, they still require physical validation and periodic renewals at counters. The limitations of these traditional methods highlight the need for a **Smart Bus Pass Management System** that automates and digitizes the entire process, making it more secure, efficient, and passenger-friendly.

3.2 Limitations of Traditional Methods:

1. **Time-Consuming Process**

- Passengers must visit ticket counters, fill out paperwork, and wait in long queues for pass issuance or renewal, leading to delays and inconvenience.

2. **Manual Verification & Risk of Fraud**

- Conductors verify passes manually, making it easy for passengers to misuse expired or fake passes, resulting in revenue loss for transport authorities

3. **Lack of Digital Integration**

   - Traditional systems do not provide online registration, digital payments, or automated renewal, requiring frequent physical visits for pass management.

4. **Paper-Based Dependency**

   - Printed passes are prone to wear and tear, loss, and forgery, increasing the need for frequent replacements and additional administrative workload.

5. **Limited Tracking & Analytics**

   - There is no centralized database to monitor pass usage, optimize routes, or analyze passenger travel patterns, making it difficult to improve public transport efficiency.

6. **Inconvenient Payment Methods**

   - Most traditional systems rely on cash payments, which can be inconvenient for passengers and create handling and accounting challenges for authorities.

7. **No Real-Time Notifications**

   - Passengers do not receive timely alerts for pass expiry, renewal, or updates, leading to last-minute rushes and travel disruptions.

8. **Lack of Accessibility**

   - Physically visiting the bus depot for pass renewal may be difficult for senior citizens, differently-abled individuals, or people with busy schedules.

CHAPTER 4 PROPOSED METHODOLOGY

4.1 Overview:

The proposed methodology for the Smart Bus Pass Management System involves leveraging digital technologies to streamline the process of issuing, renewing, and verifying bus passes. The system will consist of a web and mobile-based application where users can register, upload necessary documents, and apply for a bus pass digitally. The application will integrate a secure payment gateway for online fee payments and generate a QR-based digital pass, reducing the need for physical cards.

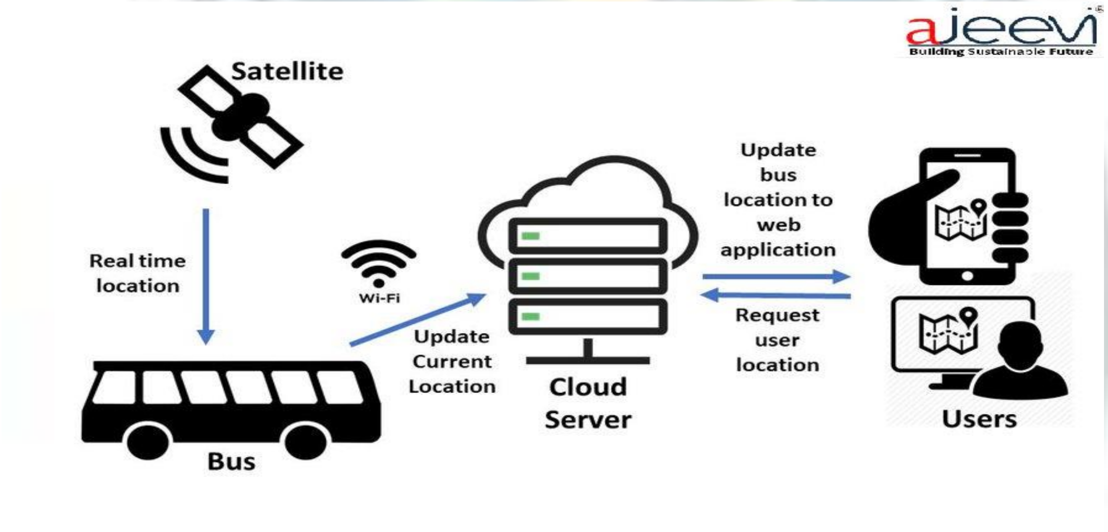


Figure 4.1: Represent Block Diagram

Step 1: Data Preparation

The system gathers data from various sources, including:

User Registration: Name, age, gender, contact details, address, ID proof.

Pass Details: Pass type (daily, weekly, monthly, student, senior citizen, etc.), validity period, fare.

Bus & Route Information: Bus numbers, routes, stops, schedules.

Transactions: Payments, top-ups, renewals, discounts.

RFID/NFC Card Data: Unique card ID, issued date, linked user account.

Step 2: Model Training and Evaluation

Problem Definition & Data Preparation: Before training a model, it's crucial to define the problem and prepare data accordingly. Some ML applications in the system include:

Passenger Demand Prediction: Forecasting the number of travelers at different times.

Fraud Detection: Identifying fake or misused bus passes.

Optimal Route Planning: Analyzing traffic patterns to recommend better bus schedules.

Fare Optimization: Dynamic pricing models based on demand.

Step 3: Prediction

Passenger Demand Forecasting : Predict the number of passengers at different times and locations to optimize bus schedules and resource allocation.

Fraud Detection in Pass Usage: Detect fraudulent activities like pass sharing, fake registrations, or unauthorized access.

Bus Route Optimization: Optimize bus routes based on passenger flow and traffic conditions.

Fare Prediction & Dynamic Pricing: Predict the best fare for different routes and times to maximize revenue and affordability.

Step 4: Existing Model

RFID/NFC-Based Smart Bus Pass Model:RFID/NFC-Based Smart Bus Pass Model.Cards are linked to user accounts and balances are updated in real-time.

QR Code-Based Digital Pass Model: Users generate a QR code on their mobile app for every trip.Scanned at the bus entry gate for fare deduction.Works with mobile wallets and UPI-based payments.

Step 5: RFID-Based Smart Bus Pass Management System

An RFID-based Smart Bus Pass Management System uses Radio Frequency Identification (RFID) technology to automate fare collection, reduce manual ticketing, and improve the efficiency of public transportation.Passengers can tap their RFID-enabled smart cards at a bus entry/exit point to deduct the fare automatically.

4.2 Data Splitting & Preprocessing

Data Splitting: In the **Smart Bus Pass Management System**, data splitting is essential for efficient processing, security, and management of user information. The data can be categorized and split into different modules based on functionality:

User Data – Includes personal details such as name, age, contact information, and user category (student, senior citizen, regular commuter). This data is stored securely and used for authentication and verification.

Bus Pass Data – Contains information related to issued passes, such as pass type, validity period, fare details, and pass status (active, expired, or pending renewal).

Data Preprocessing:

Step 1: Data Collection & Integration

Sources: Mobile apps, bus entry scanners, GPS trackers, payment gateways

Storage: MySQL, MongoDB, Firebase (for real-time data)

Integration Methods: APIs for real-time data fetching

Step 2: Data Cleaning

Raw data may have missing values, duplicates, and incorrect entries.

Techniques for Cleaning:

Handle missing values: Use mean/median for numerical, mode for categorical values

Remove duplicates: Eliminate duplicate QR scans or transactions

Correct incorrect data: Convert negative fares to absolute values

Step 3: Data Transformation

Data needs to be in a uniform format for easy processing.

Date-Time Formatting – Convert all timestamps to a standard format

Encoding Categorical Data – Convert pass types into numerical values

Normalizing Numerical Data – Scale fare amounts for better model accuracy

4.3 ML Model

The ML model implementation involves loading and sampling the dataset, then preprocessing it by encoding categorical features, scaling data, and addressing class imbalance using SMOTE. The dataset is split into training and testing sets, and models, specifically Random Forest Classifier and Extra Trees Classifier, are either trained or loaded from disk if previously saved. The trained models are used to make predictions on the test set. Performance is evaluated by calculating accuracy and generating classification reports and confusion matrices, which are then printed to assess and compare the models' effectiveness.

4.3.1 How It Works:

The Smart Bus Pass Management System automates the process of bus pass issuance, validation, and fare collection using QR codes, RFID cards, and AI-based analytics. Below is a detailed breakdown of its workflow and functionality.The Smart Bus Pass Management System automates the process of pass issuance, validation, fare collection, and real-time tracking using QR codes, RFID cards, and AI-driven analytics. Initially, users register through a mobile app, web portal, or kiosk, providing personal details and selecting their pass type. Upon successful registration and payment, a unique QR code or RFID-enabled pass is generated, which serves as their digital ticket. When boarding the bus, passengers scan their QR code or tap their RFID card on a validator, which checks the pass validity in real time through a cloud-based database.

Step 1: User Registration & Pass Issuance via a mobile app, web portal, or kiosk.

They provide personal details, ID verification, and pass type selection.

Step 2: Bus Entry & Ticket Validation When boarding, the user scans the QR code or taps the RFID card on a reader.The system validates the pass by checking its database.If the pass is valid, the system logs the trip and deducts fare (if applicable).If the pass is expired/invalid, the system alerts the driver and user.

Step 3:GPS Tracking & Real-time Monitoring The bus is tracked in real-time using GPS sensors.Users can see the estimated arrival time (ETA) in the app.

Architecture:

Web and Mobile Application: Users (passengers) can register, apply for passes, make payments, and receive a digital bus pass via a web or mobile application**.**

Bus Conductor App: A mobile app or handheld scanner allows conductors to scan the QR code and validate the pass.

Admin Dashboard: Transport authorities use a web-based dashboard to manage user data, approve/reject applications, and monitor system performance.

Disadvantages:

1. Technological Barriers

Digital Divide – Not all passengers may have access to smartphones or the internet to register and use the system.

Device Compatibility Issues – Older mobile phones may not support QR scanning or mobile apps.

2. System Reliability Issues

Network Dependency – The system relies on the internet and cloud servers; any downtime can disrupt services.

Hardware Failures – RFID scanners, QR code readers, and kiosks may malfunction, leading to validation issues.

3. Security & Privacy Concerns

Data Privacy Risks – Personal information, payment details, and travel history are stored, making the system vulnerable to hacking.

Fraud & Counterfeiting – QR codes and RFID cards can be copied or misused if proper security measures are not in place.

4. Cost & Maintenance

High Initial Investment – Setting up infrastructure like RFID readers, QR scanners, servers, and mobile apps requires significant funding.

4.3.2 Advantages:

A Smart Bus Pass Management System enhances public transportation by pass issuance, validation, and fare collection. Here are its key benefits:

1. Convenience & Efficiency

Quick Pass Issuance – Users can apply, renew, and manage bus passes online via mobile apps or web portals.

Contactless Travel – QR codes, RFID, and NFC-based validation reduce waiting times

Faster Boarding Process – Automated scanning speeds up entry, reducing delays at bus stops

2. Enhanced Security & Fraud Prevention

Unique Digital Passes – QR codes and RFID prevent duplication and misuse of bus passes.

AI-Based Fraud Detection – Detects suspicious activities such as fake or expired passes.

Secure Transactions – Encrypted payment gateways ensure safe online payments.

3. Cost Savings & Environmental Benefits

Reduces Paper Waste – Eliminates traditional paper passes, promoting eco-friendliness.

Lower Administrative Costs – Minimizes manual ticketing and reduces staff workload.

Efficient Revenue Collection – Digital payments reduce cash handling and theft risks.

4. Real-Time Tracking & Analytics

Live Bus Tracking – Passengers can track buses using GPS, improving time management.

Route Optimization – AI-based analysis helps optimize bus schedules and routes.

Passenger Insights – Data analytics provide insights into demand patterns for better planning.

5. Better User Experience

Multi-Platform Access – Users can access services via mobile apps, kiosks, and web portals.

Multiple Payment Options – Supports digital wallets, UPI, credit/debit cards, and NFC payments.

Instant Notifications – Alerts for pass expiry, bus delays, and payment updates.

CHAPTER 5

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general- purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object- oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS: The Primary goals inthe design ofthe UML are as follows:

Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

- Provide extendibilityand specialization mechanisms to extendthe core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.

5.1 Class Diagram:

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an “is-a” or “has-a” relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed “methods” of the class. Apart from this, each class may have certain “attributes” that uniquely identify the class.

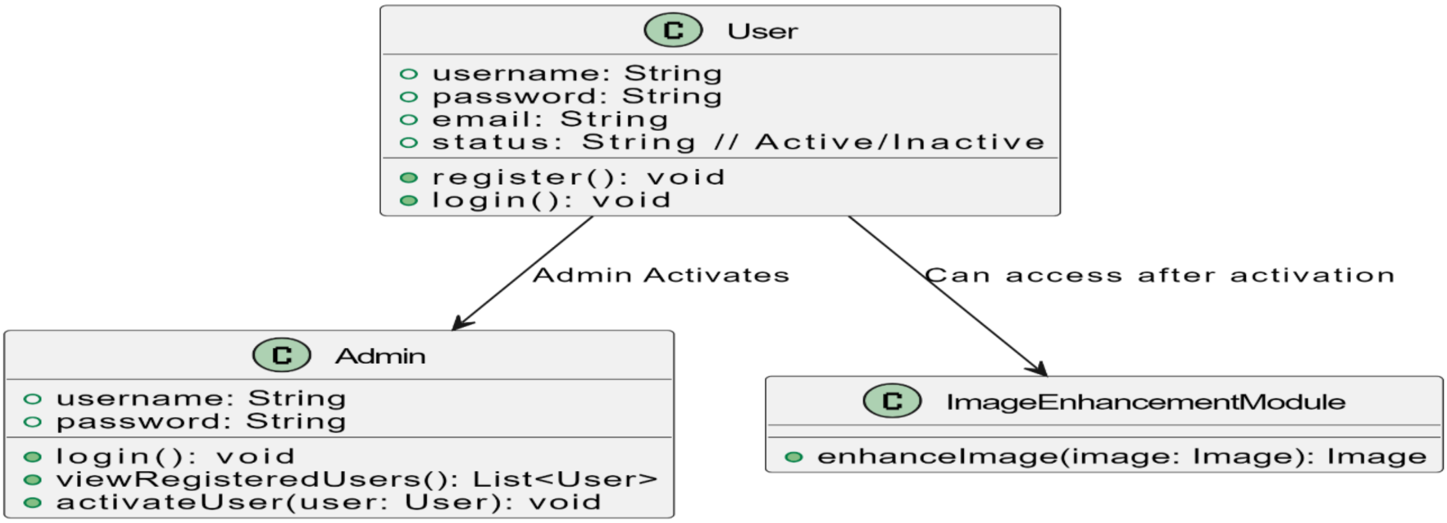


Figure 5.1: Class Diagram

5.2 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows, as parallel vertical lines (“lifelines”), different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

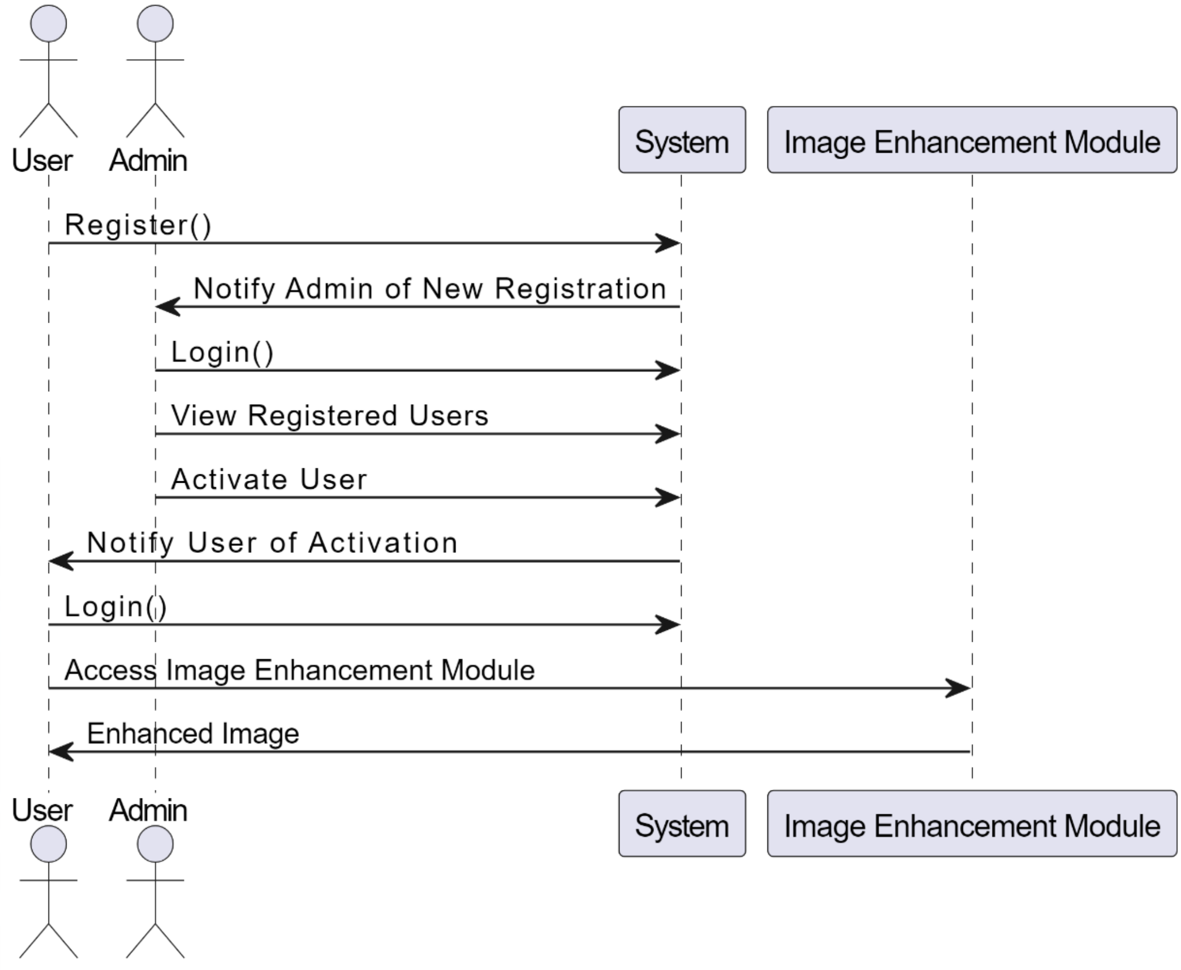


Figure 5.2: Sequence Diagram

5.3Activity diagram:

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

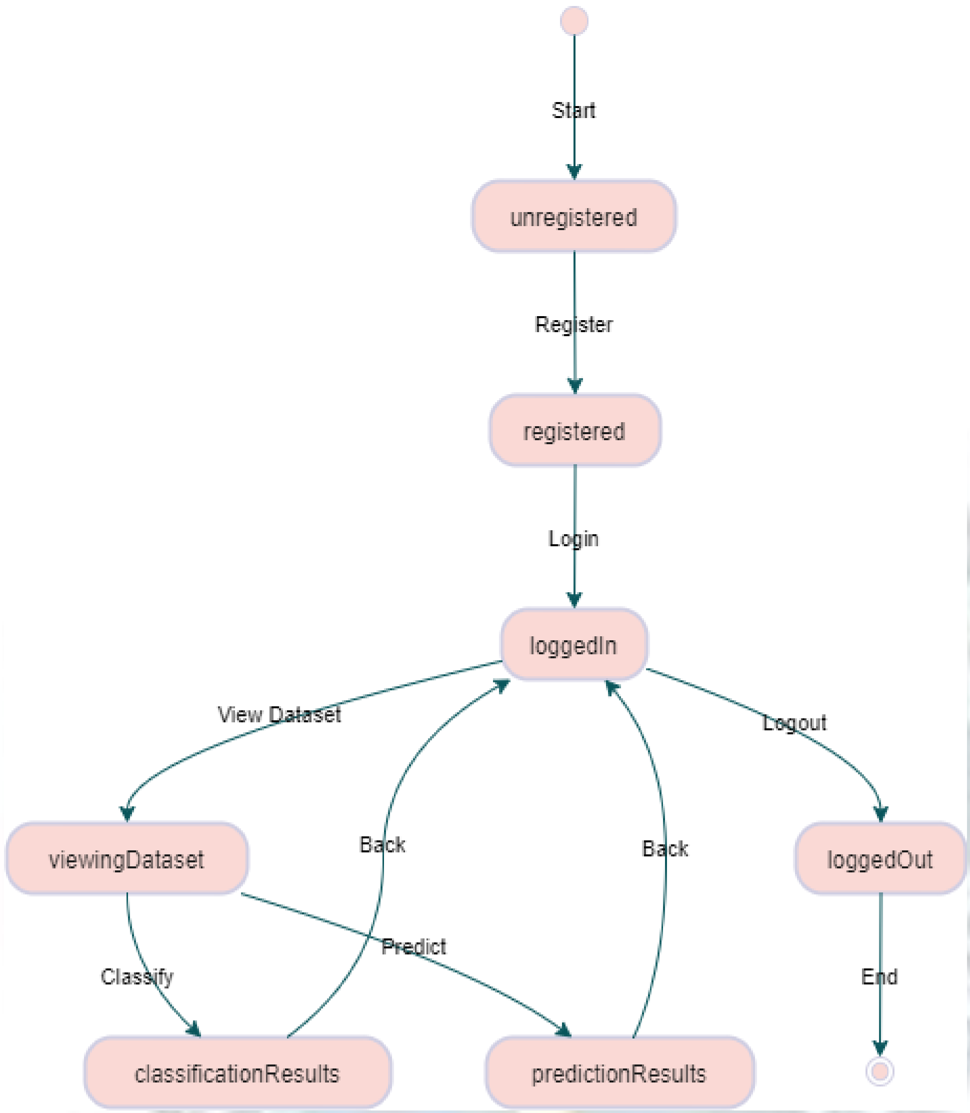


Figure 5.3: Activity Diagram

5.4 Use Case Diagram

A use case diagram in the Unified Modelling Language (UML) is a type of behavioral diagram defined by and created from a Use-Case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

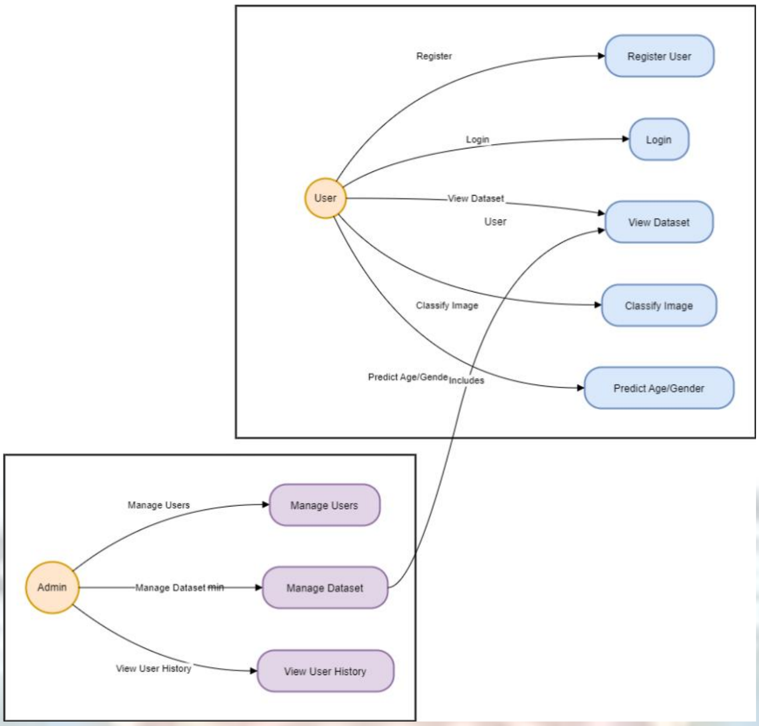


Figure 5.4: Use Case Diagram

CHAPTER 6

SOFTWARE ENVIRONMENT

6.1 What is Python?

Python is currentlythe most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.The biggest strength of Python is huge collection of standard libraries which can be used for the following :

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Imageprocessing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

6.2 Advantages of Python

1. Extensive Libraries:

   Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

2. Extensible:

   As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

1. Improved Productivity

   The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

2. IOT Opportunities

   Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

3. Simple and Easy

   When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

4. Readable

   Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

5. Object-Oriented

   This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

6. Free and Open-Source

   Like said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

7. Portable

   When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA).

Advantages of Python Over Other Languages

1. Less Coding

   Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

   Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

   The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for Everyone

   Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

6.3 Disadvantages of Python

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations

   We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

   While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle.

3. Design Restrictions

As you know, Python is dynamically-typed. This means that you don’t need to declare the type of variable while writing the code. It uses duck-typing. But wait,. While this is easy on the programmers during coding, it can raise run-time errors.

what’s that? Well, it just means that if it looks like a duck, it must be a duck

4. Underdeveloped Database Access Layers

   Compared to more widely used technologies like JDBC (Java Data Base Connectivity) and ODBC (Open Data Base Connectivity), Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

   No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

6.4 History of Python

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI).

I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it. "Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I

6.5 Python Development Steps

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.

Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python

2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.

Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x.

The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

Print is now a function.

6.6 Purpose

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

6.7 Modules Used in Project

NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities Besides its obvious scientific uses Arbitrary datatypes can be defined using NumPy which allows NumPy to seamlessly .

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Install Python Step-by-Step in Windows and Mac

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

How to Install Python on Windows and Mac

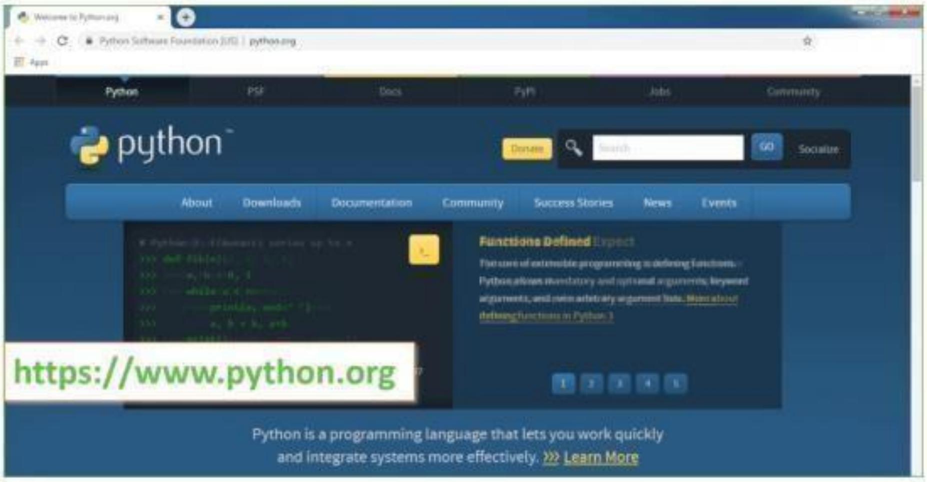
There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System RequirementsSo the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here.The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

Download the Correct version into the system

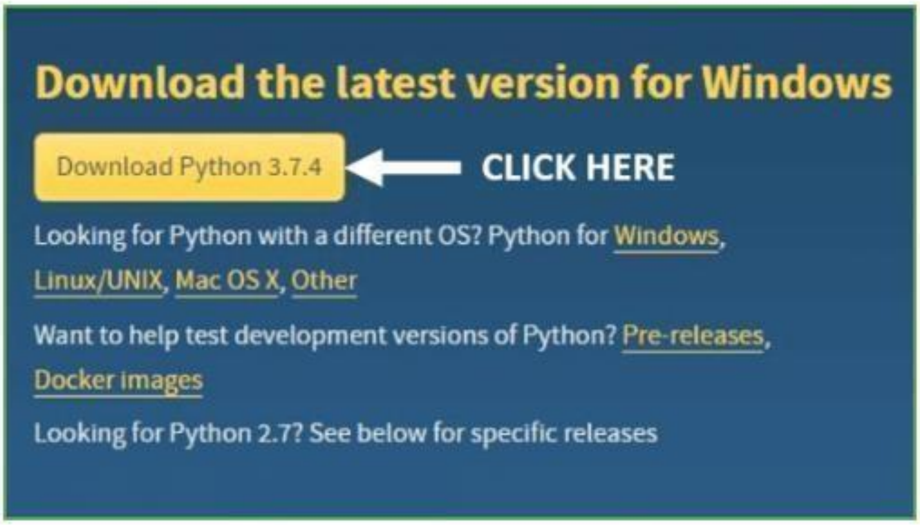
Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link:



https://www.python.org

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Now, check for the latest and the correct version for your operating system. Step 2: Click on the Download Tab.



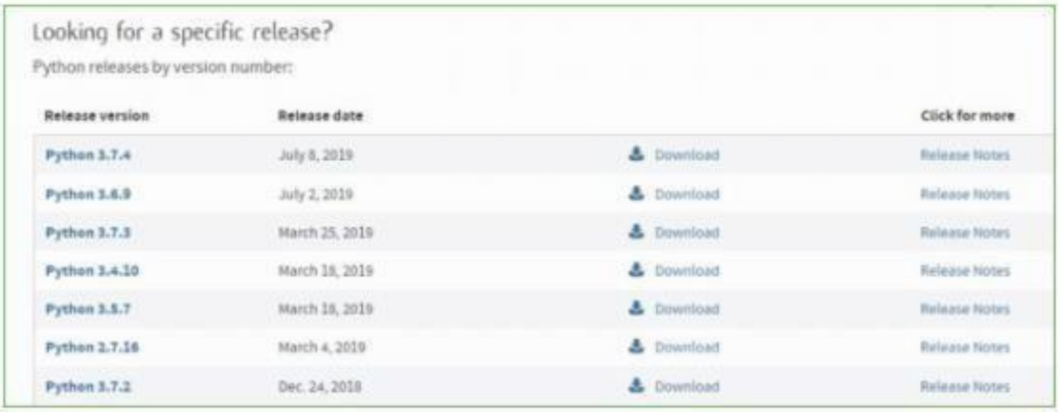
Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases, Docker images

Looking for Python 2.7? See below for specific releases

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



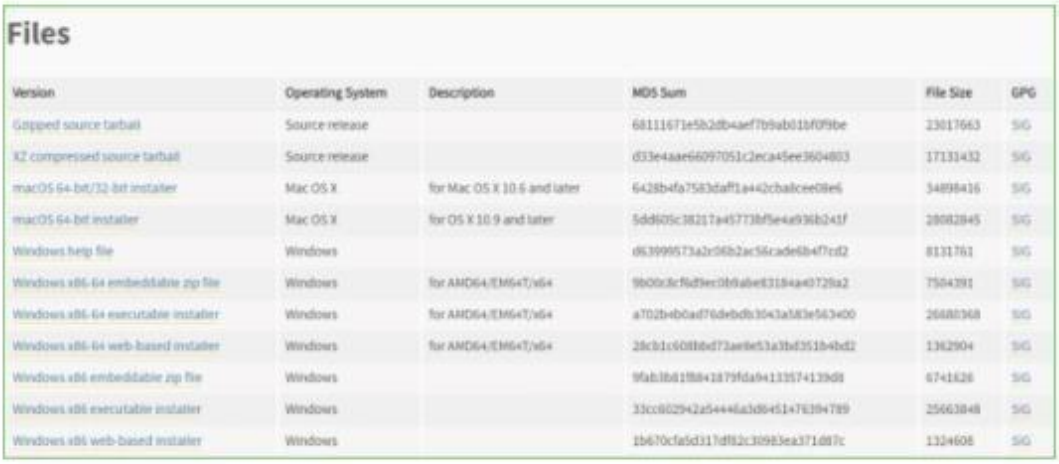
Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cba8cee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	SIG
Windows help file	Windows		d63999573a2c06b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9abe83184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76debdb3043a583e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc602942a54446a3d6451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

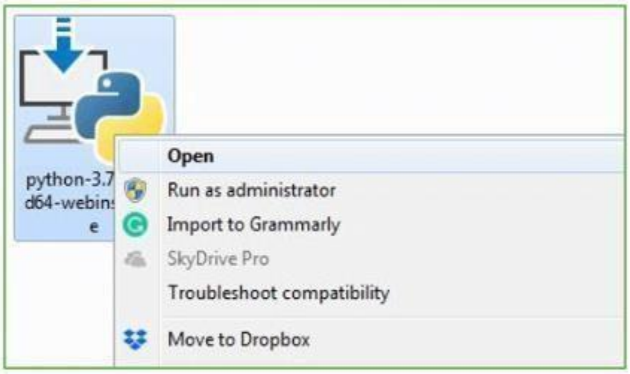
- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

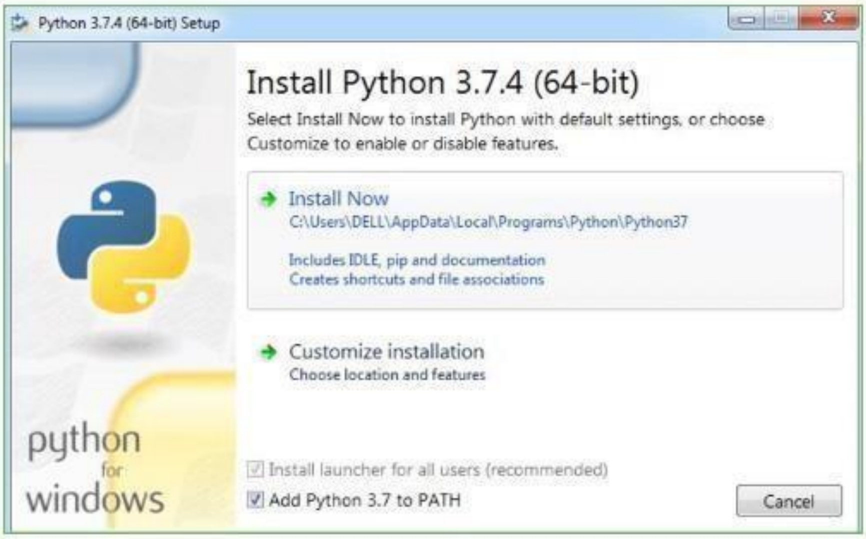
Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

6.8 Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



Step 2: Before you click on Install Now, make sure to put a tick on Add Python 3.7 to PATH.



Python 3.7.4 (64-bit) Setup

Install Python 3.7.4 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

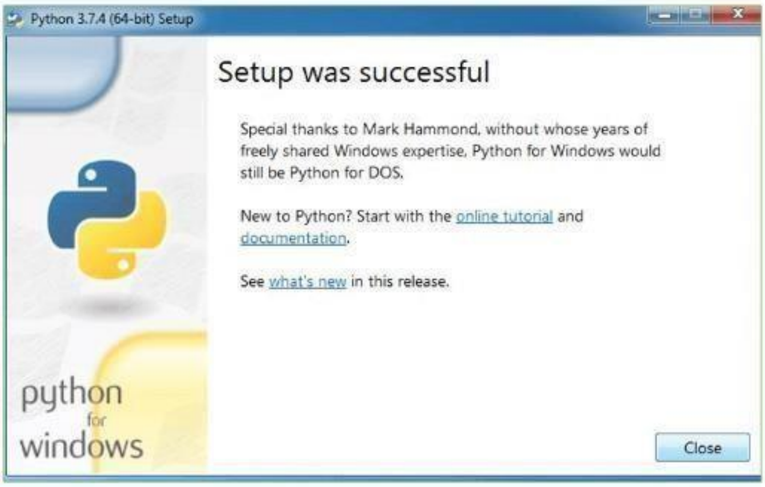
Install Now
C:\Users\DELL\AppData\Local\Programs\Python\Python37

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.7 to PATH

Step 3: Click on Install NOW After the installation is successful. Click on Close.



Python 3.7.4 (64-bit) Setup

Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

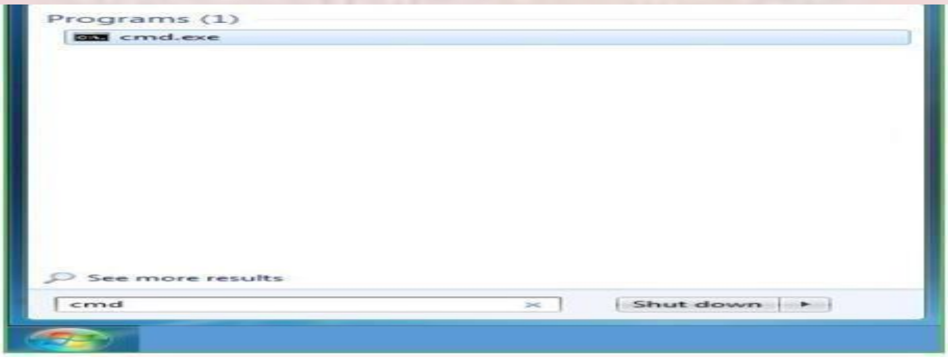
See what's new in this release.

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes. Verify the Python Installation

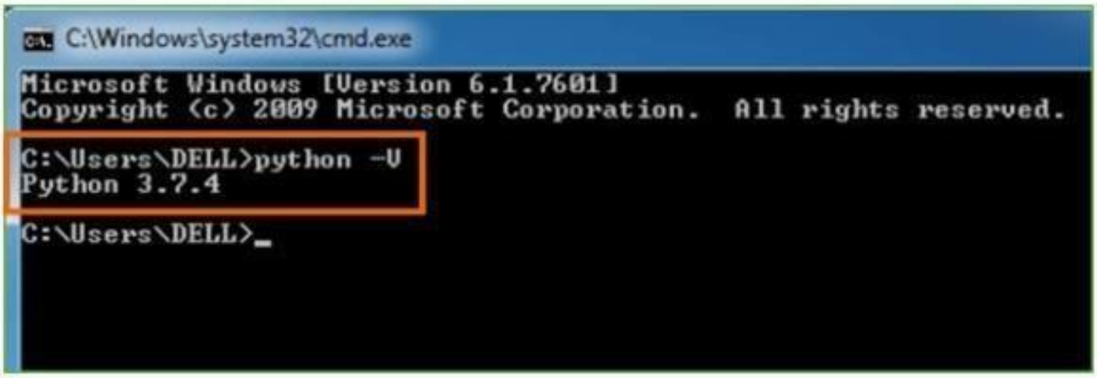
Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.



Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type python –V and press Enter.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\DELL>python -V
Python 3.7.4

C:\Users\DELL>_
```

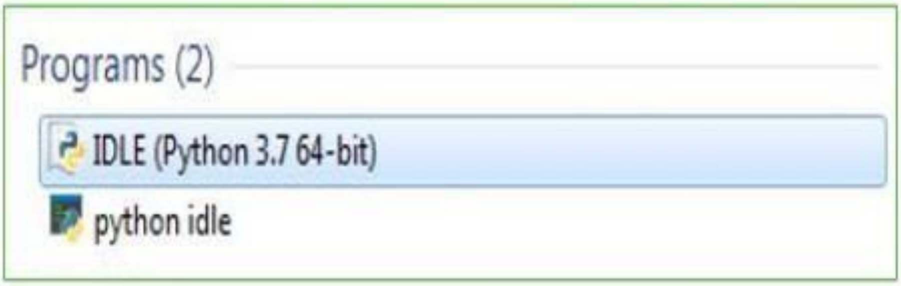
Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

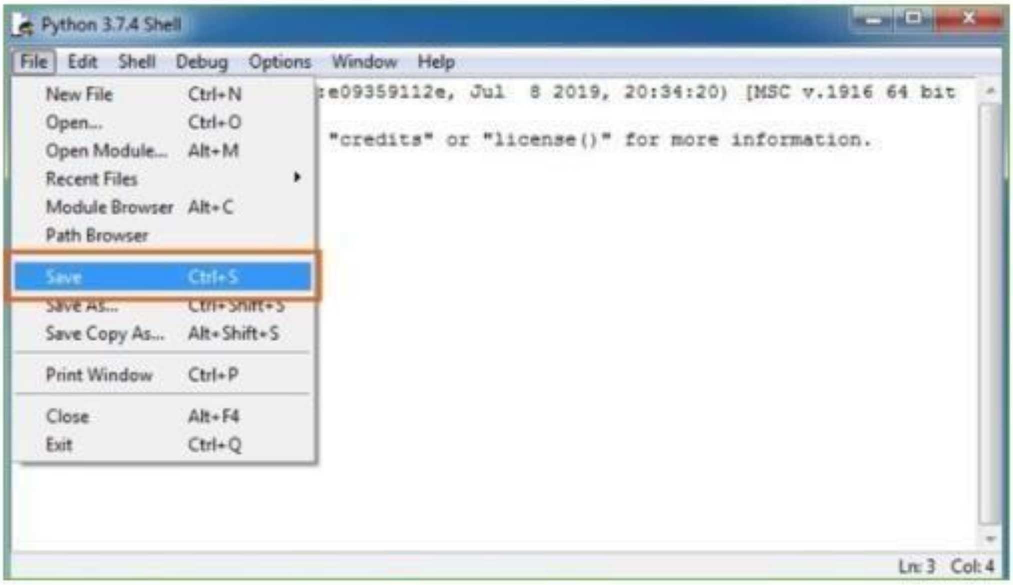
Step 1: Click on Start

Step 2: In the Windows Run



Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

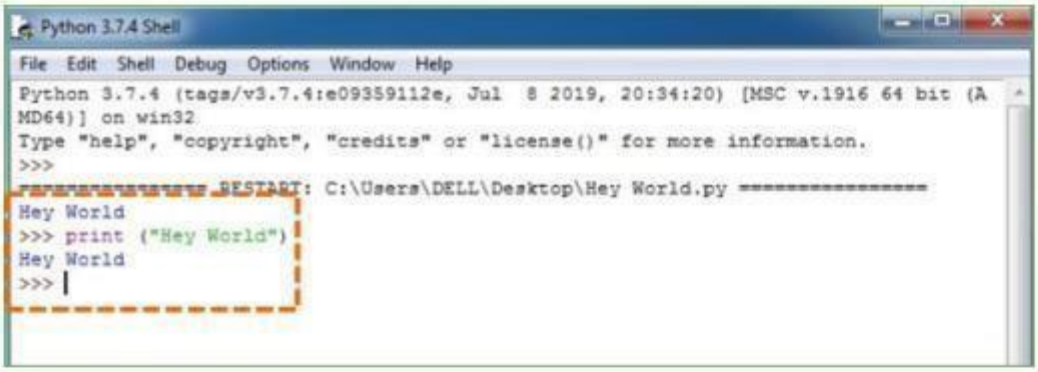
Step 4: To go ahead with working in IDLE you must first save the file. Click on File > Click on Save



```
:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit
 "credits" or "license()" for more information.
```

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. enter print (“Hey World”) and Press Enter.



```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit (A
MD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:\Users\DELL\Desktop\Hey World.py ================
Hey World
>>> print ("Hey World")
Hey World
>>> |
```

You will see that the command given is launched. With this, we end our tutorial on how to install Python. You have learned how to download python for windows into your respective operating system.

Note: Unlike Java, Python does not need semicolons at the end of the statements otherwise it won’t work.

CHAPTER 7 SYSTEM REQUIREMENTS

7.1 SOFTWARE REQUIREMENTS

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regard to what the areas of strength and deficit are and how to tackle them.

❖ **Operating system**	:	Windows 10.
❖ **Coding Language**	:	Python.
❖ **Front-End**	:	Html. CSS
❖ **Designing**	:	Html,css,javascript.
❖ **Data Base**	:	SQLite.

7.2 HARDWARE REQUIREMENTS

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

❖ **System**	:	Intel i3
❖ **Hard Disk**	:	1 TB.
❖ **Monitor**	:	14’ Colour Monitor.
❖ **Mouse**	:	Optical Mouse

CHAPTER 8

FUNCTIONAL REQUIREMENTS

OUTPUT DESIGN

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization
- Internal Outputs whose destination is within organization and they are the
- User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.

OUTPUT DEFINITION

The outputs should be defined in terms of the following points:

Type of the output

- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

INPUT DESIGN

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

INPUT STAGES

The main input stages can be listed as below:

- Data recording

- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

Input Types

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department’s communications to the system?
- Interactive, which are inputs entered during a dialogue.

Input Media

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

- Type of input
- Flexibility of format
- Speed
- Accuracy
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easy to use
- Portability

- Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive.

ERROR AVOIDANCE

At this stage care is to be taken to ensure that input data remains accurate form the stage at which it is recorded up to the stage in which the data is accepted by the system. This can be achieved only by means of careful control each time the data is handled.

ERROR DETECTION

Even though every effort is make to avoid the occurrence of errors, still a small proportion of errors is always likely to occur, these types of errors can be discovered by using validations to check the input data.

DATA VALIDATION

Procedures are designed to detect errors in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data. Whenever an invalid data is keyed in, the system immediately prompts the user and the user has to again key in the data and the system will accept the data only if the data is correct. Validations have been included where necessary.

USER INTERFACE DESIGN

It is essential to consult the system users and discuss their needs while designing the user interface:

USER INTERFACE SYSTEMS CAN BE BROADLY CLASIFIED AS:

- User initiated interface the user is in charge, controlling the progress of the user/computer dialogue. In the computer-initiated interface, the computer selects the next stage in the interaction.
- Computer initiated interfaces

In the computer-initiated interfaces the computer guides the progress of the user/computer dialogue. Information is displayed and the user response of the computer takes action or displays further information.

USER INITIATED INTERGFACES

User initiated interfaces fall into two approximate classes:

- Command driven interfaces: In this type of interface the user inputs.

- Forms oriented interface: The user calls up an image of the form to his/her screen and fills in the form. The forms-oriented interface is chosen because it is the best choice.

COMPUTER-INITIATED INTERFACES

The following computer – initiated interfaces were used:

- The menu system for the user is presented with a list of alternatives and the user chooses one; of alternatives.
- Questions – answer type dialog system where the computer asks question and takes action based on the basis of the users reply.

Right from the start the system is going to be menu driven, the opening menu displays the available options. Choosing one option gives another popup menu with more options. In this way every option leads the users to data entry form where the user can key in the data.

ERROR MESSAGE DESIGN

The design of error messages is an important part of the user interface design. As user is bound to commit some errors or other while designing a system the system should be designed to be helpful by providing the user with information regarding the error he/she has committed.

This application must be able to produce output at different modules for different inputs.

PERFORMANCE REQUIREMENTS

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

CHAPTER 9

SOURCECODE

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bootstrap Page with Sidebar</title>
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
    <style>
        body {
            background: linear-gradient(45deg, #ff6b6b, #ffcc33);
            font-family: 'Comic Sans MS', cursive, sans-serif;
            color: #fff;
            text-align: center;
            padding: 20px;
        }


sidebar {
            height: 100vh;
            background-color: #343a40; /* Background color for the sidebar */
            padding-top: 20px;
            position: fixed;
            width: 250px;
        }
        .sidebar .nav-link {
            color: #fff; /* Text color for the sidebar links */
            margin: 10px 0;
        }
        .sidebar .nav-link:hover {
            color: #d4d4d4; /* Hover color for the sidebar links */
        }
        .content {
            margin-left: 270px; /* Adjust this based on the sidebar width */
            padding: 20px;
        }
 </style>
</head>
<body>
    <div class="sidebar">
        <a class="navbar-brand text-center d-block mb-4" href="#">Sleep Disorders</a>
        <ul class="nav flex-column">



       <li class="nav-item">
```

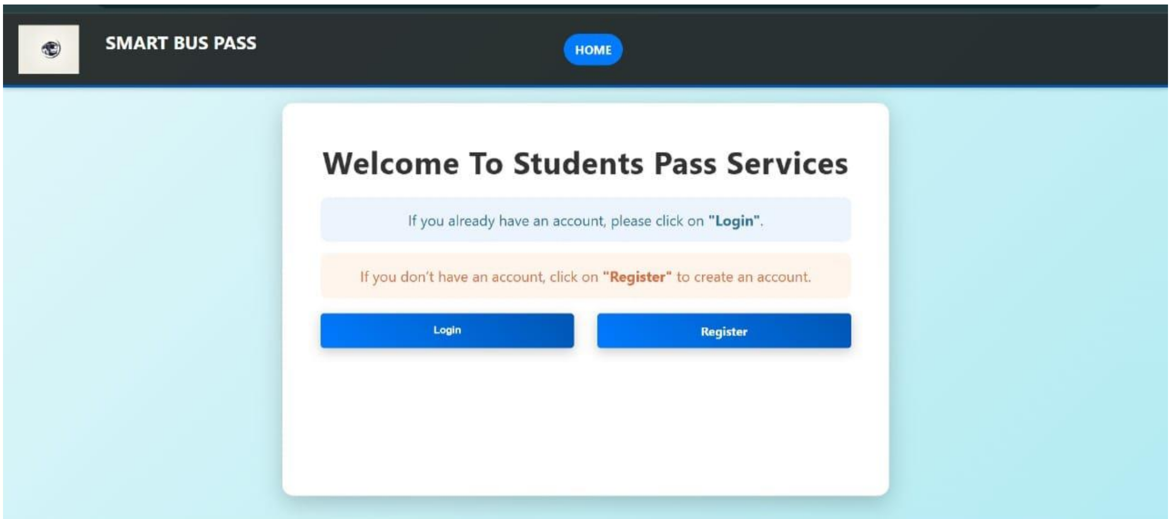
```
                <a class="nav-link" href="{% url 'index' %}">Home</a>
            </li>
            <li class="nav-item">

    <a class="nav-link" href="{% url 'userLogin' %}">User-Login</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="{% url 'adminLogin' %}">Admin-Login</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="{% url 'userregister' %}">User-Registration</a>
            </li>
        </ul>
    </div>
        {% block contents %}
        {% endblock %}

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>


            </div>
            <div class="service-item">
              <div class="icon">📊</div>
              <h3>Data Visualization</h3>
              <p>Clear and insightful data representation for better understanding.</p>
            </div>
            <div class="service-item">
              <div class="icon">🤖</div>
              <h3>AI-Powered Predictions</h3>
              <p>Accurate predictions to support timely interventions.</p>
            </div>
          </div>
        </section>


        <section id="about" class="about">
          <div class="container">
            <h3>About Us</h3>
            <p>Our deep learning system helps farmers identify crop stress early, enabling
      timely intervention and preventing significant yield losses.</p>
```

```
        <ul>
          <li>Utilizes advanced deep learning models for accurate analysis.</li>
          <li>Trained on a large and diverse dataset of crop images.</li>
          <li>Provides actionable insights to improve crop health and yields.</li>
        </ul>
      </div>
    </section>
  </main>

  <footer>
    <p>© 2024 MSL Corporations. All Rights Reserved.</p>
  </footer>
</body>

</html>

{% load static %}

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">
  <title>Crop Stress Detection Project - Admin Panel</title>
  <link rel="stylesheet" type="text/css" href="{% static 'style.css' %}">

</head>

<body>
 <header class="header">
   <div class="container">
     <h1 class="sitename">Crop Stress Detection</h1>
     <nav class="navmenu">
       <ul>
         <li><a href="userHome" class="active">Home</a></li>
        <li><a href="Classification_result">Classification_result</a></li>
        <li><a href="prediction">prediction</a></li>

        <li><a href="#about">About Us</a></li>
        <li><a href="#featured-services">Services</a></li>
        <li><a href="/">Logout</a></li>

      </ul>
    </nav>
  </div>
</header>
<div class="space"></div>
<main>
  <section class="hero">
    <div class="img-box">
      <img src="templates/static/hero.jpg" alt="">
    </div>
      <div class="hero-content">
        {% block hero %}

        {% endblock %}
      </div>
    </div>

  </section>
  <section id="featured-services" class="featured-services">
    <div class="container">
      <div class="service-item">
        <div class="icon">🌾</div>
        <h3>Stress Detection</h3>
        <p>Identify various types of crop stress (water, nutrient, disease etc.).</p>
      </div>
      <div class="service-item">
        <div class="icon">📸</div>
        <h3>Image Analysis</h3>
        <p>Analyze images of crops to detect stress indicators.</p>
        </div>
        <div class="service-item">
          <div class="icon">📊</div>
          <h3>Data Visualization</h3>
          <p>Clear and insightful data representation for better understanding.</p>
        </div>
        <div class="service-item">
          <div class="icon">🤖</div>
          <h3>AI-Powered Predictions</h3>
          <p>Accurate predictions to support timely interventions.</p>
        </div>
      </div>
    </section>

    <section id="about" class="about">
      <div class="container">
        <h3>About Us</h3>
        <p>Our deep learning system helps farmers identify crop stress early, enabling timely intervention and preventing significant yield losses.</p>
        <ul>
          <li>Utilizes advanced deep learning models for accurate analysis.</li>
          <li>Trained on a large and diverse dataset of crop images.</li>
          <li>Provides actionable insights to improve crop health and yields.</li>
        </ul>
      </div>
    </section>
  </main>

  <footer>
    <p>© 2024 MSL Corporations. All Rights Reserved.</p>
  </footer>
</body>

</html>


from django.shortcuts import render
from Users.utility.classification_report import Classification_report
from .models import UserRegistrationModel
from django.contrib import messages

# Create your views here.

def register(request):
    if request.method=='POST':
        name=request.POST['name']
        email=request.POST['email']
        loginid=request.POST['loginid']
        password=request.POST['password']
        mobile=request.POST['mobile']
        locality=request.POST['locality']
        state=request.POST['state']

        try:

            userRegister=UserRegistrationModel(name=name,email=email,loginid=loginid,password=password,mobile=mobile,locality=locality,address=state)
            if userRegister:
                userRegister.save()
            else:
                messages.error(request,'Invalid Details , Enter deatils carefully')
                return render(request,'userRegister.html')
        except Exception as e:
            pass
            messages.error(request,e)
            return render(request,'userRegister.html')

def userLoginCheck(request):
    if request.method=="POST":
        loginid=request.POST['username']
        password=request.POST['password']


        try:
            user=UserRegistrationModel.objects.get(loginid=loginid,password=password)

            status=user.status
            print(status)
            if status=='activated':

                return render(request,'users/userHome.html')
            else:
                messages.error(request,'Status Not Actiavted')
                return render(request,'userLogin.html')
        except Exception as e:
            pass
            messages.error(request,'Invalid details')
            return render(request,'userLogin.html')

def Classification_result(request):
    accuracy=Classification_report()
    return render(request,'users/classification_view.html',{'accuracy':accuracy})

from django.core.files.storage import default_storage
from .utility.prediction import predict_image

def prediction(request):
    if request.method == 'POST' and 'image' in request.FILES:
        # Save uploaded image temporarily
        image_file = request.FILES['image']
        file_path = default_storage.save('temp/' + image_file.name, image_file)
        temp_image_path = default_storage.path(file_path)
        predicted_label=predict_image(temp_image_path)

        return render(request,'users/prediction.html',{'predicted_label':predicted_label})
    return render(request,'users/prediction.html')
import os
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from sklearn.metrics import confusion_matrix, classification_report,
ConfusionMatrixDisplay

# Suppress warnings
import warnings
warnings.filterwarnings("ignore")

def create_data_generators(train_dir, validation_dir, test_dir, img_size=224,
batch_size=32):
    """Create ImageDataGenerators for training, validation, and testing."""
    train_datagen = ImageDataGenerator(
        rescale=1.0 / 255,  # Normalize pixel values between 0 and 1
        rotation_range=20,  # Augment images with slight rotation
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True,
    )

    validation_datagen = ImageDataGenerator(rescale=1.0 / 255)
    test_datagen = ImageDataGenerator(rescale=1.0 / 255)

    train_gen = train_datagen.flow_from_directory(
        train_dir,
        target_size=(img_size, img_size),
        batch_size=batch_size,
        class_mode="categorical",
    )

    validation_gen = validation_datagen.flow_from_directory(
        validation_dir,
        target_size=(img_size, img_size),
        batch_size=batch_size,
        class_mode="categorical",
    )

    test_gen = test_datagen.flow_from_directory(
        test_dir,
        target_size=(img_size, img_size),
        batch_size=batch_size,
        class_mode="categorical",
        shuffle=False,
    )

    return train_gen, validation_gen, test_gen

def build_cnn_model(img_size=224, num_classes=2):
    """Build the CNN model."""
    model = Sequential([
        Conv2D(32, (3, 3), activation="relu", input_shape=(img_size, img_size, 3)),
        MaxPooling2D(2, 2),
        Conv2D(64, (3, 3), activation="relu"),
        MaxPooling2D(2, 2),
        Conv2D(128, (3, 3), activation="relu"),
        MaxPooling2D(2, 2),
        Conv2D(256, (3, 3), activation="relu"),
        MaxPooling2D(2, 2),
        Flatten(),
        Dense(512, activation="relu"),
        Dropout(0.5),
        Dense(num_classes, activation="softmax"),
    ])
    return model

def compile_and_train_model(model, train_gen, validation_gen, epochs=40):
    """Compile and train the model with callbacks."""
    model.compile(optimizer="adam", loss="categorical_crossentropy",
metrics=["accuracy"])

    # Callbacks
    early_stopping = EarlyStopping(monitor="val_loss", patience=5, restore_best_weights=True)
    reduce_lr = ReduceLROnPlateau(monitor="val_loss", factor=0.2, patience=3, min_lr=1e-6)

    history = model.fit(
        train_gen,
        validation_data=validation_gen,
        epochs=epochs,
        callbacks=[early_stopping, reduce_lr],
    )

    return model, history

def evaluate_and_visualize(model, test_gen, history):
    """Evaluate the model and visualize results."""
    # Evaluate the model
    test_loss, test_accuracy = model.evaluate(test_gen)
    print(f"\nTest Loss: {test_loss:.3f}")
    print(f"Test Accuracy: {test_accuracy:.3f}")

    # Plot accuracy and loss curves
    acc = history.history["accuracy"]
    val_acc = history.history["val_accuracy"]
    loss = history.history["loss"]
    val_loss = history.history["val_loss"]
    epochs_range = range(len(acc))

    plt.figure()
    plt.plot(epochs_range, acc, label="Training Accuracy")
    plt.plot(epochs_range, val_acc, label="Validation Accuracy")
    plt.title("Training and Validation Accuracy")
    plt.legend()

    plt.figure()
    plt.plot(epochs_range, loss, label="Training Loss")
    plt.plot(epochs_range, val_loss, label="Validation Loss")
    plt.title("Training and Validation Loss")
    plt.legend()
    plt.show()

   def generate_confusion_matrix(model, test_gen):
    """Generate confusion matrix and classification report."""
    predictions = model.predict(test_gen)
    predicted_classes = np.argmax(predictions, axis=1)
    true_classes = test_gen.classes
    class_labels = list(test_gen.class_indices.keys())

    conf_matrix = confusion_matrix(true_classes, predicted_classes)
    disp = ConfusionMatrixDisplay(confusion_matrix=conf_matrix,
display_labels=class_labels)
    disp.plot(cmap=plt.cm.Blues)
    plt.title("Confusion Matrix")
    plt.show()

    print("\nClassification Report:")
    print(classification_report(true_classes, predicted_classes, target_names=class_labels))

def Classification_report(img_size=224, batch_size=32, epochs=40):
    train_dir = r'D:\data-
point\Crop_stress_analysis_and_detection_using_deep_learning\media\Dataset\train'
    validation_dir = r'D:\data-
point\Crop_stress_analysis_and_detection_using_deep_learning\media\Dataset\validation'
    test_dir = r'D:\data-
point\Crop_stress_analysis_and_detection_using_deep_learning\media\Dataset\test'
    model_save_path = r'D:\data-
point\Crop_stress_analysis_and_detection_using_deep_learning\media\Dataset\crop_stres
s_model.h5'
    """Main function to run the entire model training and evaluation pipeline."""
    # Step 1: Prepare data generators
    train_gen, validation_gen, test_gen = create_data_generators(train_dir, validation_dir,
test_dir, img_size, batch_size)


    # Display class indices
    print("\nClass indices:", train_gen.class_indices)
    # Step 2: Build and compile the model
    model = build_cnn_model(img_size, len(train_gen.class_indices))

    # Print the model summary
    model.summary()

    # Step 3: Train the model
    model, history = compile_and_train_model(model, train_gen, validation_gen, epochs)

    # Step 4: Save the model
    if model_save_path:
        model.save(model_save_path)
        print(f"\nModel saved at {model_save_path}")

    # Step 5: Evaluate and visualize the results
    evaluate_and_visualize(model, test_gen, history)

    # Step 6: Generate confusion matrix and classification report
    generate_confusion_matrix(model, test_gen)

    accuracy= model.evaluate(test_gen)[1]  # Return test accuracy

    return accuracy
```

```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import load_img, img_to_array
import matplotlib.pyplot as plt

# Load the trained model (update the path to your model)


def preprocess_image(image_path, img_size):
    """
    Preprocesses the image for prediction.
    - Resizes to `img_size`.
    - Converts to array and normalizes pixel values.
    """
    image = load_img(image_path, target_size=(img_size, img_size))
    image_array = img_to_array(image)
    image_array = np.expand_dims(image_array, axis=0)
    image_array = image_array / 255.0
    return image_array,image

def predict_image(image_path):
    model_save_path = r'D:\data-point\Crop_stress_analysis_and_detection_using_deep_learningg\CODE\Crop_stress_analysis_and_detection_using_deep_learning\media\Dataset\crop_stress_model.h5'
    model = load_model(model_save_path)

# Class labels (update with your actual labels used during training)
    class_labels = ['BLB','BLAST','HEALTHY','HISPA','LEAP_SPOT']
    """
    Predicts the class of the preprocessed image.
    - Returns the predicted class label and confidence.
    """
    image_array,image=preprocess_image(image_path,img_size=224)
    predictions = model.predict(image_array)
    predicted_class_index = np.argmax(predictions)
    predicted_class_label = class_labels[predicted_class_index]
    confidence = predictions[0][predicted_class_index]

    print(f"Predicted Class Index: {predicted_class_index}")
    print(f"Predicted Class Label: {predicted_class_label}")
    print(f"Prediction Confidence: {predictions[0][predicted_class_index]:.2f}")
    plt.imshow(image)
    plt.title(f"Prediction: {predicted_class_label}")
    plt.axis("off")
    plt.show()
   return predicted_class_label
```

CHAPTER 10

EXPERIMENTAL RESULTS

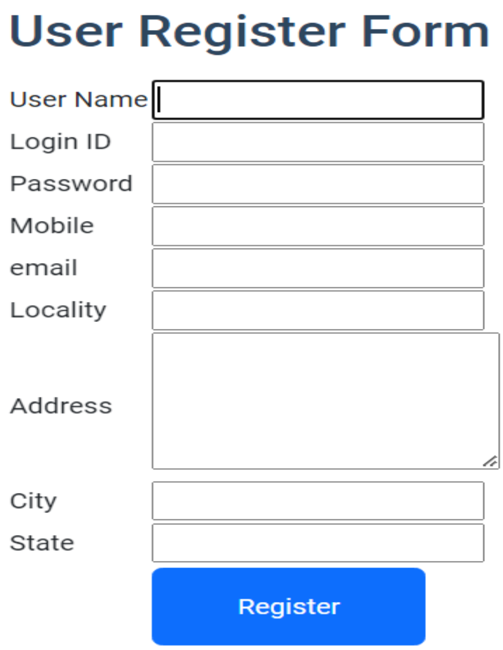


Welcome To Students Pass Services

If you already have an account, please click on "**Login**".

If you don't have an account, click on "**Register**" to create an account.

Fig 10.1:Index Page



User Register Form

User Name

Login ID

Password

Mobile

email

Locality

Address

City

State

Fig 10.2: User Register Page

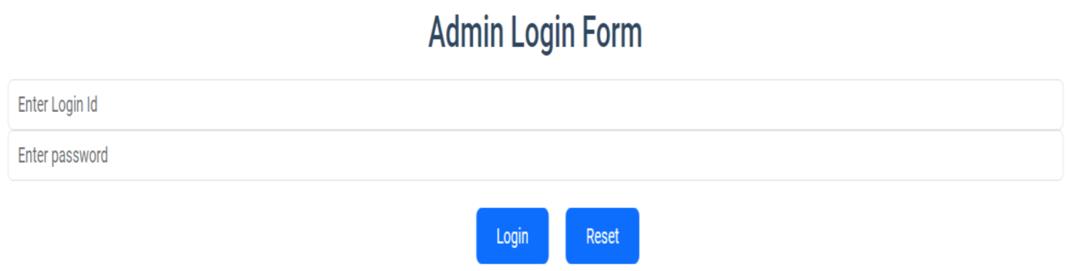


Fig 10.3:Admin Login Page

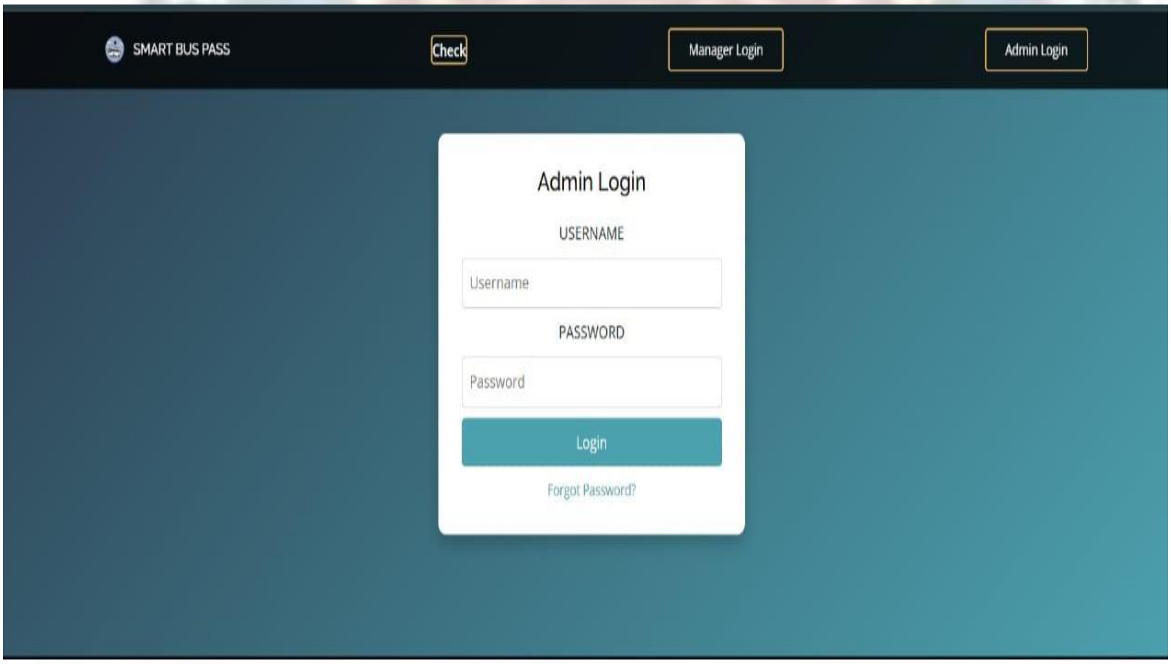
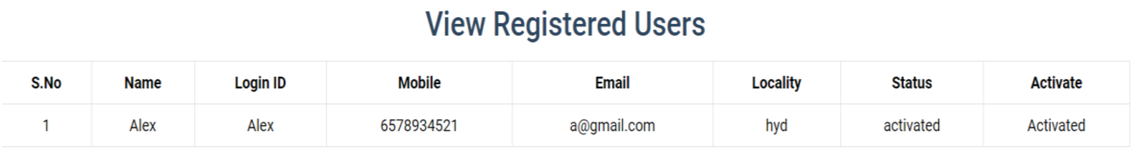


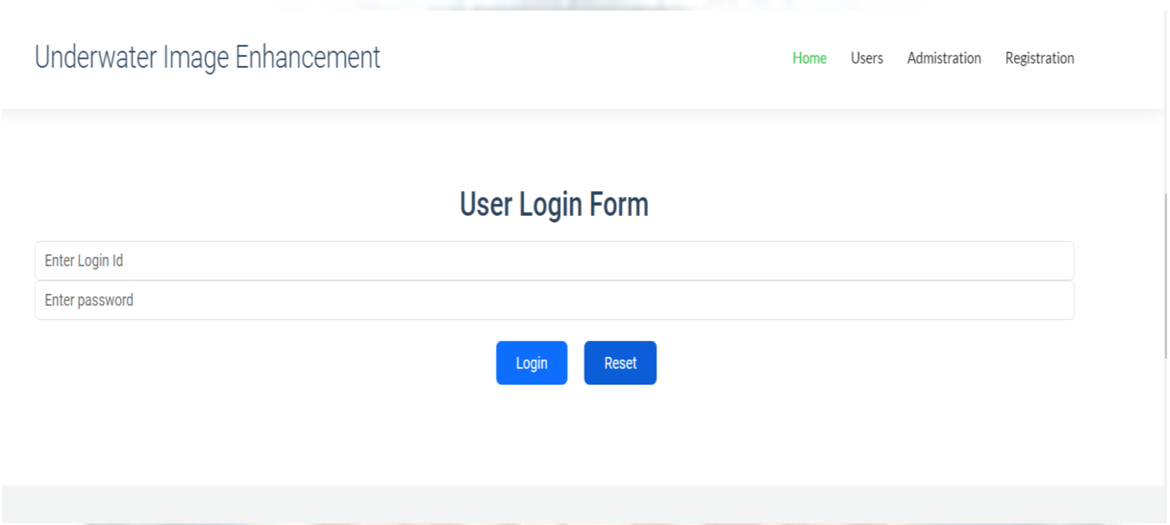
Fig 10.4: Admin Page



View Registered Users

S.No	Name	Login ID	Mobile	Email	Locality	Status	Activate
1	Alex	Alex	6578934521	a@gmail.com	hyd	activated	Activated

Fig 10.5: Admin/ User details Panel



Underwater Image Enhancement

User Login Form

Enter Login Id

Enter password

Fig 10.6: User Login Form

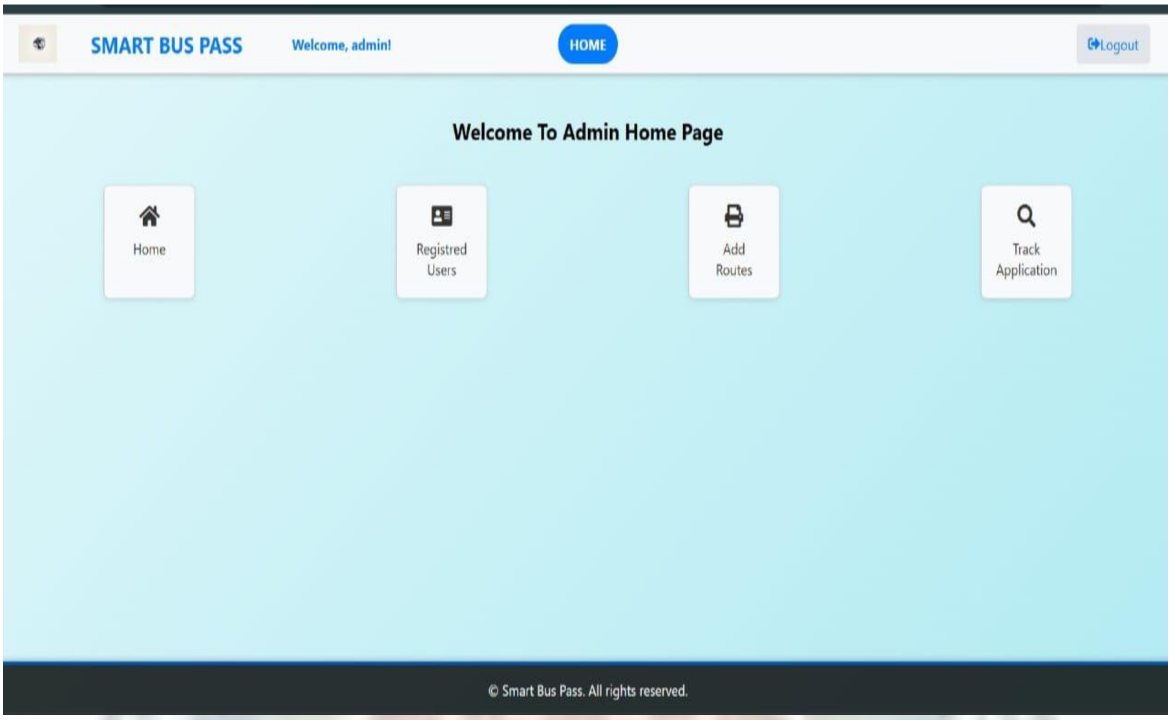
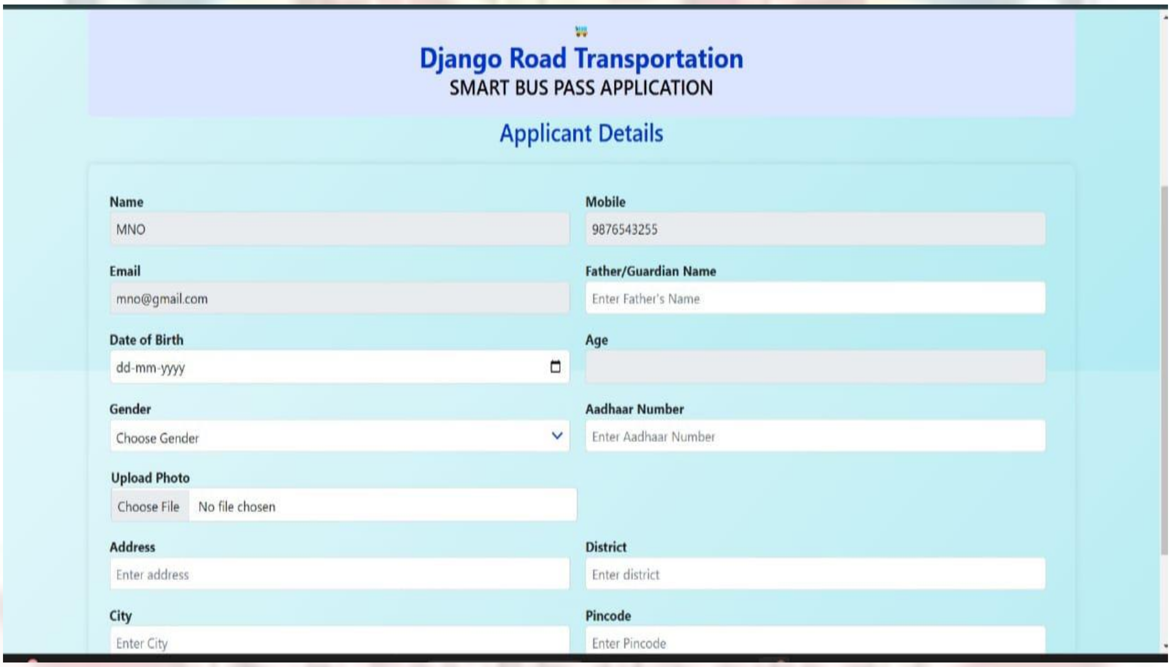


Fig 10.7: Admin Home Page



Name: MNO

Mobile: 9876543255

Email: mno@gmail.com

Father/Guardian Name: Enter Father's Name

Date of Birth: dd-mm-yyyy

Age:

Gender: Choose Gender

Aadhaar Number: Enter Aadhaar Number

Upload Photo: Choose File No file chosen

Address: Enter address

District: Enter district

City: Enter City

Pincode: Enter Pincode

Fig 10.8: Application Details

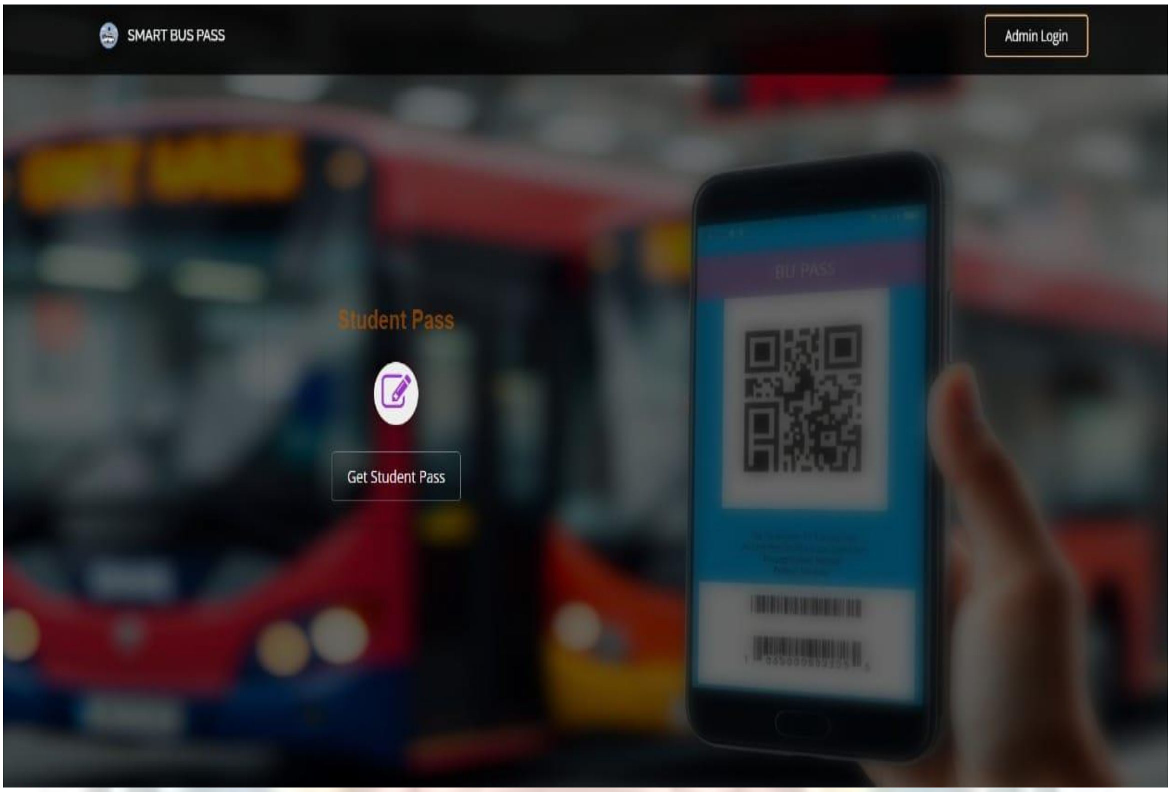


Fig 10.9 : User/Prediction Page

CHAPTER 11

CNOCLUSION AND FUTURE SCOPE

Conclusion

The Smart Bus Pass Management System represents a significant advancement in the modernization and digital transformation of public transportation services. With the increasing demand for efficient, eco-friendly, and user-centric transportation solutions, this system emerges as a comprehensive tool to enhance the overall bus commuting experience for both passengers and operators. By integrating modern technologies such as Radio Frequency Identification (RFID), Near Field Communication (NFC), and mobile applications, the Smart Bus Pass Management System not only streamlines ticketing processes but also brings greater transparency, security, and accessibility to public transport services..

The system is built on the principle of making public transport more efficient, reducing human intervention, and eliminating the need for traditional paper-based tickets, which are often prone to damage, loss, or fraud. This is accomplished through the use of digital passes, which can be stored and accessed on smartphones or RFID-enabled cards. Users simply tap their smart devices or cards on the bus reader to gain access to services, reducing the time spent in queues, and improving the overall efficiency of the boarding process. For passengers, this represents a significant convenience, as they no longer need to carry physical tickets or worry about managing cash during travel. Instead, the digital pass is always with them, whether on a smartphone app or an RFID card, ready to be used for multiple One of the primary benefits of the Smart Bus Pass Management System is the reduction of administrative overhead and operational costs. Traditional methods of issuing physical tickets or passes can be cumbersome, expensive, and prone to error. Paper tickets are not only costly to print and distribute but also require additional resources to manage and track. With a smart system in place, ticketing is automated, reducing the need for manual intervention and making the entire process smoother and more cost-effective. Furthermore, as more passengers adopt the system, there is a reduction in the need for cash transactions, which eliminates the possibility of fraud and theft that often occurs in cash-handling processes. From an operational perspective, the Smart Bus Pass Management System also allows for real-time tracking and monitoring of passenger movements.

Transport authorities can gather valuable data regarding ridership patterns, peak hours, and frequent routes.

Future Scope

The future scope of this project encompasses several key areas for expansion and enhancement:

The future scope of the Smart Bus Pass Management System is vast and holds immense potential for transforming public transportation systems globally. As technology continues to advance, the system can evolve to integrate emerging technologies such as artificial intelligence (AI) and machine learning (ML) for predictive analytics, enabling real-time dynamic route optimization and personalized travel experiences based on passenger behavior. The system could also expand to include multi-modal transport integration, allowing seamless connectivity between buses, trains, metros, and other public transport services, providing a comprehensive mobility solution.

Smart Bus Pass Analysis

The Smart Bus Pass Management System is a transformative solution that streamlines public transportation by replacing traditional paper tickets with digital passes, utilizing technologies like RFID and mobile apps. This system enhances operational efficiency, reduces administrative costs, and provides greater convenience for passengers, enabling seamless, contactless travel. It also offers real-time data collection, allowing transport authorities to monitor passenger behavior, optimize routes, and improve service quality. The integration of secure, digital payment methods helps reduce fraud and eliminates cash handling

Architecture

The learning architecture for a Smart Bus Pass Management System integrates several key components to optimize the entire transportation experience. It begins with data collection through RFID-enabled cards or mobile apps, capturing real-time information on passenger entry, bus locations, and payment transactions This data is then processed and stored in a centralized database, where machine learning algorithms are applied for predictive analytics, route optimization, and demand forecasting. Supervised learning models predict passenger demand, adjust bus schedules, and optimize routes based on traffic and historical patterns, while reinforcement learning continuously refines real-time decision-making for dynamic route adjustments. Data security and user authentication are prioritized through encryption and secure .

Augmented Training Data

Smart Bus Pass Management System involves enhancing the dataset used to train machine learning models by incorporating additional, synthetic, or varied data points to improve the system's performance and robustness. This could include simulating different passenger behaviors, such as varying travel times, peak and off-peak usage patterns, and fare payment methods across different regions or times of day. For example, the data could be augmented by introducing synthetic scenarios like higher-than-usual passenger traffic during holidays, severe weather conditions affecting bus schedules, or even unusual traffic patterns due to accidents or roadwork. Additionally, augmented data could involve introducing variations in user demographics, such as age, disabilities, or different levels of digital literacy, to ensure the system is inclusive and adaptable to diverse user needs.

Fusion Methods

Smart Bus Pass Management System involve combining multiple data sources and models to enhance decision-making, improve accuracy, and provide a more comprehensive understanding of passenger behavior and operational efficiency. These methods can include **data fusion**, where information from various sensors (e.g., RFID readers, GPS, and mobile apps) is integrated to create a unified view of the system's performance. For instance, data from bus locations, passenger movements, and payment transactions can be fused to provide real-time insights into bus congestion, travel times, and potential delays, allowing for dynamic route optimization. Additionally, **model fusion** can be used, combining multiple machine learning models to leverage their individual strength

User-Friendly Decision Support Systems

user-friendly decision-making approach for a Smart Bus Pass Management System focuses on making the system intuitive, accessible, and easy to navigate for passengers while ensuring smooth operations for transport authorities. For passengers, the system should prioritize simplicity in the user interface, offering mobile apps or web portals that provide quick access to purchasing, managing, and using bus passes. Clear instructions, easily readable text, and a few simple steps for ticket purchase or pass activation are essential for reducing complexity. Features like automatic trip history tracking, route suggestions based on travel habits, and real-time bus tracking.

Digital Bus Pass Generation

Digital bus pass generation for a Smart Bus Pass Management System revolutionizes the traditional ticketing process by offering a convenient, secure, and eco-friendly way for passengers to access public transport. Instead of relying on physical paper tickets, passengers can generate a digital bus pass through a mobile app or web platform. Upon registration, users can input their personal details, select their preferred pass type (daily, monthly, or yearly), and make secure online payments. Once the transaction is complete, the system generates a unique, encrypted digital pass stored on the user's smartphone or a smart .The digital pass is then linked to the user’s account.

Smart Bus Pass Renewal and Expiry Alerts

Smart Bus Pass renewal and expiry alerts are essential features of a Smart Bus Pass Management System, designed to enhance user convenience and ensure uninterrupted travel. As the expiration date of a bus pass approaches, the system sends timely notifications to users via mobile apps, emails, or SMS alerts, reminding them to renew their pass before it expires. These alerts can be personalized based on the type of pass—whether daily, monthly, or yearly—providing users with sufficient time to complete the renewal process..

Tracking The Routes

Tracking routes in a Smart Bus Pass Management System plays a crucial role in ensuring efficient operations, timely arrivals, and a seamless passenger experience. Real-time route tracking is enabled through GPS technology embedded in buses, allowing both transportation authorities and passengers to monitor the location and progress of buses at any given time. For passengers, this feature provides real-time updates on bus arrival times, allowing them to plan their trips more effectively and reducing wait times at bus stops.

Real Time Analytics

Real-time analytics for a Smart Bus Pass Management System plays a pivotal role in improving the overall efficiency, user experience, and operational decision-making. It involves the continuous collection, processing, and analysis of data as it happens, allowing both passengers and transport authorities to make informed decisions instantly

QR Code and RFID-based verification

It enhances both the security and efficiency of passenger access to public transportation. These two technologies streamline the process of boarding buses while ensuring secure and accurate validation of digital bus pass. The reader scans the tag and verifies the user's pass in real-time through a central server.

Security And Access Control

These are vital components of a Smart Bus Pass Management System to ensure that user data is protected, services are secure, and the system functions efficiently without being vulnerable to fraud or unauthorized access. The implementation of robust security measures and access control mechanisms helps safeguard both the passengers' personal information and the integrity of the entire transportation system..

CHAPTER 12 REFERENCE

[1] M. Furkhan and H. R. Divakar, “Smart Bus Pass System Using Android”, IJRESM, vol. 5, no. 7, pp. 71–74, Jul. 2022.

[2] S.Famitha, G, “Online Bus pass Generation System using Web Application ”2019.

[3] Prof. N. V. Chaudhari, Komal Malewar, Komal Sukhadeve, Jayashri Uke. Arti Panpate, "Android Application for Issuing Bus Pass" Dr. Babasaheb Ambedkar College of Engineering and Research Nagpur, India, 2020

[4] Patel, Bhumik, and Parthvi Pandey. "RFID Based Bus Ticketing system.", International journal of scientific research and engineering trends. Volume 4 issue 2, mar-apr (2018).

[5] Hu, N., Wei, G., Jihui, M., Design and Implementation of Bus Monitoring System Based on GPS for Beijing Olympics International Journal of Engineering Trends, Vol 7, No4, pp.540 544,2017.

[6] Sridevi.K1, Jeevitha.A2, Kavitha.K3, Narmadha.K, Sathya.K., Smart Bus Tracking and Management System using IOT, International journal of advanced engineering technology, vol 5, pp.453-458,2017.

[7] Singh, A., & Kapoor, S. (2019). "Intelligent bus ticketing system using mobile applications and IoT." *International Journal of Advanced Research in Computer Science*, 10(3), 42-49.

[8] Bajaj, A., & Soni, A. (2018). "Public transport management and ticketing system based on RFID and mobile apps." *International Journal of Innovative Research in Computer Science and Technology*, 6(3), 15-22.