**Internship Report**

**Submitted in partial fulfillment for**

**The award of the degree of**

**Master of Science**

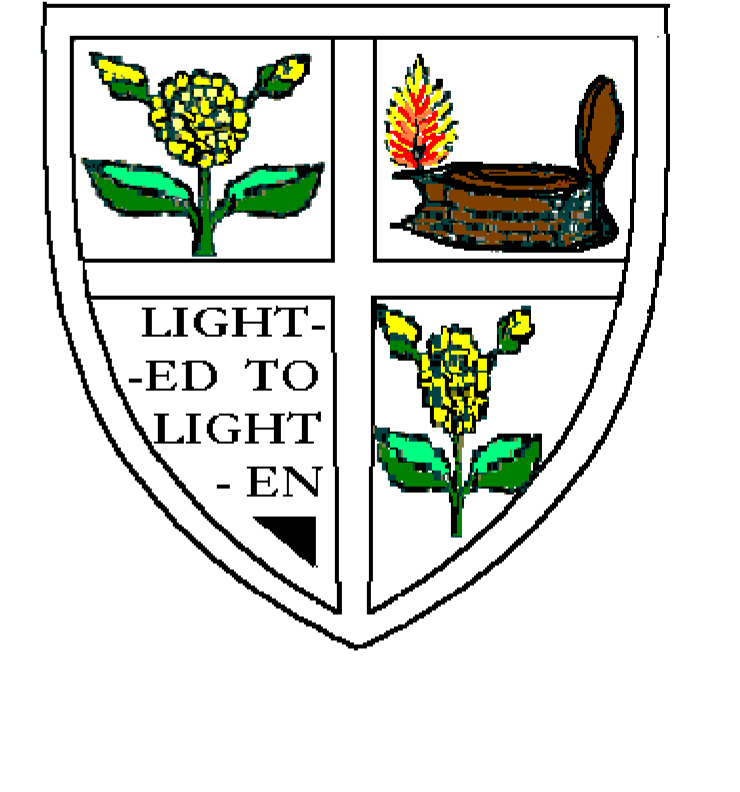
**in**

**Data Science**

**By**

**RIYA THOMAS.N**

**(23PDS14)**



**Department of Computer Science**

**Data Science**

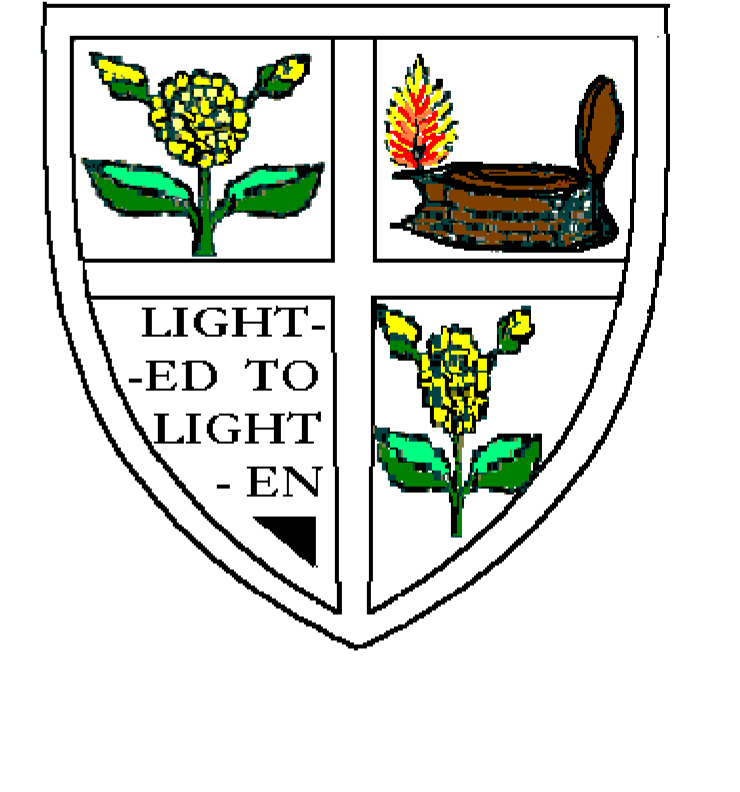
**Women’s Christian College, Chennai-06.**

**AUGUST 2024**

**WOMEN’S CHRISTIAN COLLEGE**

**DEPARTMENT OF COMPUTER SCIENCE**

**DATA SCIENCE**



**Bonafide Certificate**

**Register Number: 23PDS14**

This is to certify that the Internship Report is a bonafied work doneby **RIYA THOMAS.N**,

**(23PDS14)**, submitted to the Department of Data Science, Women’s Christian College, in

partial fulfilment of the requirement of the award of the degree of Master of Science in Data

Science.

Submitted for the viva-voce examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_ at Women’s Christian

College, Chennai.

Internal Examiner External Examiner \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**COMPANY BONAFIDE CERTIFICATE**



**DECLARATION**

I hereby declare that the Internship Report submitted to the Department of Data Science, Women’s Christian College, Chennai – 600006 is a Bonafied work done by **RIYA THOMAS.N**, **23PDS14**, submitted to the Department of Data Science, Women’s Christian College and this reportwork has not formed the basis or submitted for the award of any Degree/ Diploma or any similar title. I further certify that this work did not form a part of any other published work.

Place: Chennai

Date: 28/08/2023 Signature of the Candidate

**ACKNOWLEDGEMENT**

I would like to take this opportunity to express my deepest gratitude to the individuals and organizations that have been instrumental in making my internship at TRIPLEM INFOTECH SOLUTIONS PRIVATE LTD a valuable and enriching experience.

First and foremost, I would like to extend my heartfelt thanks to Mr.Muhammed Ilyas, for providing me with the opportunity to be a part TRIPLEM INFOTECH SOLUTIONS PRIVATE LTD of and for being an exceptional mentor throughout my internship journey. Your guidance, support, and unwavering commitment to my professional growth have been truly invaluable.

I am also grateful to the entire Team at TRIPLEM INFOTECH SOLUTIONS PRIVATE LTD for their warm welcome, support, and willingness to share their knowledge and expertise. Working alongside such dedicated and talented professionals has been an honor, and I have learned immensely from each one of you.

I would like to express my gratitude to the Principal and the Professors, academic advisors at Women’s Christian College, whose guidance and mentorship prepared me for this internship and helped me connect theoretical knowledge to practical application.

Lastly, I would like to thank my family and friends for their unwavering support and encouragement throughout this journey. Your belief in me has been a constant source of motivation.

I am deeply thankful to everyone who has played a part in making my internship at TRIPLEM INFOTECH SOLUTIONS PRIVATE LTD a memorable and enriching experience. The knowledge and skills I have acquired during this internship will undoubtedly shape my future endeavors.

Sincerely,

RIYA THOMAS.N

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **INDEX** | **PAGE NO** |
| 1 | Synopis | 7 |
| 2 | Organization Profile | 15 |
| 3 | Company Domain | 18 |
| 4 | Usage of Computer | 21 |
| 5 | Role in the Company | 24 |
| 6 | Learning during the Internship | 29 |
| 7 | Conclusion | 67 |
| 8 | Web References | 68 |

1. **SYNOPSIS**

**PROJECT TITLES:**

* Website Prediction using different datasets
* Weather app
* QR code Generator
* Google Translator

**1.1 Introduction:**

The advent of the internet has revolutionized the way we interact with information, services, and each other. In this digital age, web applications serve as the cornerstone of online experiences, offering dynamic, interactive, and personalized content to users across the globe. From ecommerce platforms and social networks to productivity tools and entertainment services, web applications power a diverse array of functionalities and serve a multitude of purposes.

At the heart of many web applications lies Flask, a Python microframework renowned for its simplicity, flexibility, and extensibility. Flask empowers developers to build web applications quickly and efficiently, leveraging the rich ecosystem of Python libraries and tools while providing the freedom to design and structure applications according to specific requirements.In this synopsis, we delve into the process of developing a web application using Flask, coupled with HTML, CSS, and database connectivity. By combining these technologies, developers can create robust, scalable, and user-friendly web applications tailored to meet the needs of modern users and businesses.

The journey begins with setting up the development environment, configuring Flask, and establishing a solid foundation for building the application. With Flask in place, attention turns to designing the database schema, defining the structure of the application's data model, and ensuring efficient storage, retrieval, and manipulation of data. Backend development takes center stage as routes are defined, view functions implemented, and business logic encapsulated within the Flask application. These backend components handle incoming HTTP requests, interact with the database, and orchestrate the flow of data and logic to deliver dynamic content and services to users. On the frontend, HTML templates come to life with the help of Flask's templating engine, Jinja2, enabling the generation of dynamic content and seamless integration with backend data. CSS style sheets add a layer of polish and sophistication, enhancing the visual appeal and usability of the web application across different devices and screen sizes.

As the development process unfolds, considerations for performance optimization, security enhancements, testing and debugging, and continuous integration and deployment (CI/CD) come to the fore. These aspects ensure that the web application meets the highest standards of quality, reliability, and user experience, driving its success and longevity in the competitive landscape of the web.

In conclusion, the journey of developing a web application with Flask is not just about writing code—it's about crafting immersive digital experiences, solving real-world problems, and connecting with users in meaningful ways. With Flask as our guide and HTML, CSS, and database connectivity as our companions, we embark on a journey of innovation, creativity, and collaboration, shaping the future of the web one line of code at a time.

**1.2 Objectives:**

To gain hands-on experience in developing a web-based application that leverages machine learning for predictive analytics in an educational context, enhancing my technical skills in **web development, machine learning integration, and user interface design.** The primary goal of this project is to create a functional prototype of a student mark prediction system that demonstrates the practical application of AI and web technologies.

**1.2.1 TECHNICAL IMPLEMENTATION:** Flask for the frontend and Browser DB as the backend framework for prediction. Let’s make sure you have Flask and Browser DB installed in your environment: For that u need to install pip install flask in pycharm terminal and u need install pip install dB browser

**1.2.2 USER EXPERIENCE ENCHANCEMENT:** Our main objective is to provide users with an easy-to-use analysis tool on a web page, minimizing complexity. This tool allows users to effortlessly input data and receive predictions, reducing time complexity and enhancing the overall user experience. We aim to achieve this by offering a clean, intuitive interface that is both accessible and responsive.

**1.2.3 DATA INTEGRATION:** Integrate input data from users into the machine learning model to provide real-time predictions. Ensure seamless data flow between the frontend and backend components of the application.

**1.2.4 SET UP ENVIRONMENT:** To set up your Flask development environment in PyCharm, begin by downloading and installing PyCharm from the Jet Brains website, then create a new project within PyCharm, specifying the project location and selecting the Python interpreter while enabling the option to create a virtual environment. Next, install Flask using pip within the virtual environment created for your project. Configure PyCharm to recognize the virtual environment as the project interpreter, and create Python files for your Flask application, along with that create a folder templates that help us return the render templates And Optionally, install Flask-SQLAlchemy for database connectivity and define database models within your Flask application for interaction with the database using SQLAlchemy's ORM. This structured approach ensures a seamless setup for developing Flask web applications in PyCharm.

To explore the fusion of machine learning techniques with data analysis for effective prediction and analysis of our dataset in a web framework in a local browser. This project focuses on utilizing Flask and database connectivity to implement machine learning techniques for analyzing a student dataset, aiming to enhance educational insights and decision-making processes. The primary objective is to develop a web application using Flask, a Python web framework, integrated with database connectivity to effectively manage and analyze student data. The project aims to apply machine learning algorithms to predict student performance and identify key factors influencing academic success. By incorporating Flask and database connectivity, the web application will enable real-time access to student data analysis results, providing valuable insights for educators and administrators to tailor interventions and support mechanisms. This project underscores the practical application of machine learning, data analysis, and web development technologies in the education sector, facilitating data-driven decision-making and student success initiatives.

**1.3 Methodology:**

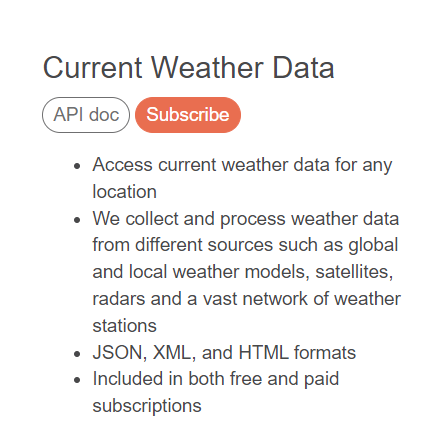
The methodology for a dataset prediction project involves a structured approach to ensure accurate and reliable results. Here’s a step-by-step methodology for implementing a dataset prediction system:

**1.3.1 Website prediction using different dataset in flask:**

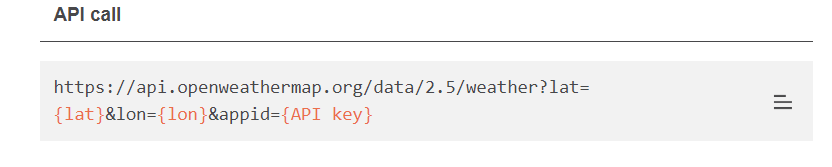
* Gather requirements from stakeholders to understand their expectations and constraints in order for to predict we need to identify the dependent variable and independent variable.
* Identify and gather relevant data from various sources e.g. from kaggle, GitHub, GoogleLLC, Ensure data is collected in a format suitable for analysis (e.g., CSV, JSON).
* Normalize or standardize numerical features encode categorical variables using techniques like one-hot encoding or label encoding.
* Split the dataset into training, validation, and test sets to ensure the model generalizes well to unseen data. Typically, a common split is 80% training, 20 % testing.
* Perform EDA to understand the data distribution, relationships, and patterns .Visualize data using plots like bar charts, boxplot, and pie chart.
* Choose appropriate machine learning algorithms based on the nature of the prediction task (e.g., regression, classification).Consider multiple algorithms to compare their performance.
* Serialize the trained model using libraries like 'joblib' or pickle for later use in the application.

**1.3.2 Weather App:**

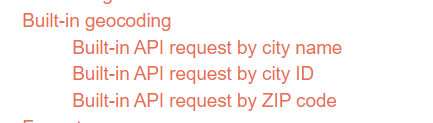
* **Python:** Ensure Python is installed on your system. You can download it from python.org
* **Tkinter:** Ensure Tkinter is installed in your python interpreter if not install using the command **pip install Tkinter**
* **Additional Library requests:** This library is used to make HTTP requests. Install it using **pip install request**
* The first step is to Importing Required Libraries
* **from tkinter import \***
* **from tkinter import ttk**
* **import requests**
* tkinter is the standard Python interface to the Tk GUI toolkit.
* ttk is a submodule of tkinter which provides themed widget set.
* Requests is used to handle HTTP requests to the weather API.
* After importing the necessary libraries next step is to fetch the weather data using <https://openweathermap.org>
* Create a personal account and create registration to get the API KEYS
* Now to forecast the weather in each city u need to go in openweather and click current weather data like the below mention



* Now u need to click the API doc



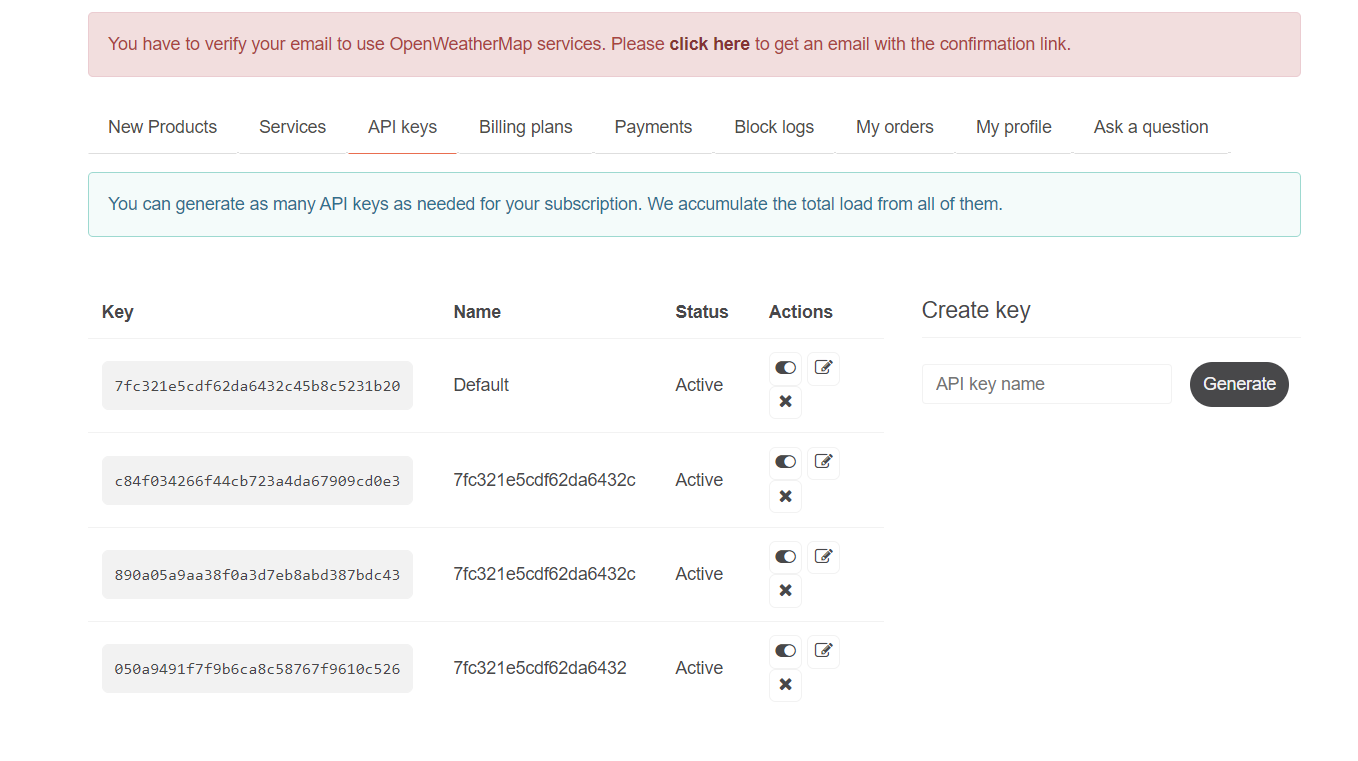
* After getting the API call link now you need to select the city\_name API to forecast weather prediction for each city.



* Here you need to select the Build-in API request by city name.



* Then you will get actual API call for city name.
* After you need to select the API key that will provide you with many API keys which might help you to forecast the weather prediction as u do in Google you might get results accurately



* Here ensure u choose correct API KEY
* We now create a data\_get() function
* data\_get(): This function is triggered when the "DONE" button is pressed.
* city\_name.get(): Retrieves the city name from the combo box.
* requests.get(): Sends a request to the OpenWeatherMap API to get weather data for the specified city.
* **title():** Sets the title of the window.
* **config():** Configures the background color of the window.
* **geometry():** Sets the dimensions of the window.
* **Adding Widgets to the WindowLabel():** Creates a label widget.
* **place():** Positions the widget at specified coordinates with specific dimensions.

Now we need to create a ComboBox for city selection for this you need to import ttk:

* **ttk.Combobox ():** Creates a combo box widget with the list of state names.
* **StringVar ():** Creates a string variable to store the selected city name.
* **Button ():** Creates a button widget.
* **command =data\_get:** Associates the data\_get() function with the button click event.

**1.3.3 QR code generator:**

* Install the QR code Library to generate QR code
* PIP INSTALL qrcode
* qr.make("https://wcc.edu.in/"): This function generates a QR code for the given URL. It creates an image object containing the QR code

.

* img.save("wcc\_qr\_code.png"): This saves the QR code image as a PNG file named "wcc\_qr\_code.png" in the current working directory.

**1.3.3 Google Translator:**

* **Python:** Ensure Python is installed on your system. You can download it from python.org
* **Tkinter:** Ensure Tkinter is installed in your python interpreter if not install using the command **pip install Tkinter**
* Create a function that translates the languages from english to hindi
* Create a googletranslator to support various languages
* Create a combobox
* Here we are creating a tkinter and create a window for gui
* We create labels
* Here we create text fields
* Here we also give command to trigger the event and translate the language

**2. ORGANIZATIONAL PROFILE:**

**2.1 Overview:**

**Name of the Organization:** TripleM Infotech Solutions Pvt Ltd



**TripleM Infotech Solutions Pvt Ltd** is a Chennai based Web Development, Web Design, Mobile App Development, Search Engine Optimization and leading web and marketing solution provider in India. We offers the full range of online marketing services with proper auditing of website performance and build apt Website Design and Website Development with SEO marketing plans with suitable strategies for your business that cover International and Geographic target audience to drag the visitors into clients, this build your brand value in your target market. For every client, we provide free Premium Website Auditing and consulting for SEO, based on that we trigger your Website towards target audience or we rebuild your Website more interactive and Responsive Website that helps to convert business leads.

**Foundation:** TripleM Infotech Solutions Pvt Ltd is a dynamic player in the tech industry, based in Chennai, India. Incorporated on September 7, 2010, it operates as a private limited company with a focus on web development, design, mobile app development, and search engine optimization.

**2.2 Mission Statement:**

“Gain valuable work experience.”

The company prides itself on offering a comprehensive suite of services that cater to the digital needs of its clients. From crafting static and dynamic websites to developing responsive e-commerce solutions, TripleM Infotech ensures that businesses make a significant online impact. Their expertise extends to digital marketing, where they assist startups in branding and online presence, ensuring that companies register effectively with government bodies.

TripleM Infotech also specializes in hosting services, managing web applications, and emails on their cloud server, providing a range of packages to suit various business needs. Their commitment to innovation is evident in their offerings of SEO, SEM, SMM, and on-demand online training through their e-learning portal.

**2.3 Core Values:**

This company deals with many IT projects and they also train us with different fields like:

* Data science
* Python programming
* Androids
* Cloud computing
* Cyber security

TRIPLEM INFOTECH Pvt Ltd is a premier Information Technology company renowned for its diverse range of IT services and comprehensive training programs. The company excels in delivering cutting-edge solutions tailored to meet the unique needs of businesses across various industries. With a strong focus on innovation and quality, TRIPLEM INFOTECH offers custom software development, creating bespoke applications that enhance operational efficiency and performance. Their web development services are designed to build responsive and user-friendly websites that drive user engagement and business growth. Additionally, the company specializes in mobile application development, crafting robust and scalable apps for both Android and iOS platforms to meet the demands of the mobile-first world

Cloud solutions are another cornerstone of TRIPLEM INFOTECH's offerings, providing businesses with optimized operations, enhanced data security, and scalable resources through cloud-based technologies. Beyond their extensive project development capabilities, TRIPLEM INFOTECH is also a hub for learning and professional development, offering training in various high-demand fields. Their training programs cover Data Science, Python Programming, Android Development, Cloud Computing, and Cyber Security, equipping individuals with the skills needed to thrive in the tech industry. These programs are designed to provide hands-on experience and deep insights into each field, ensuring participants are well-prepared for the challenges of the digital world.

In the realm of Data Science, TRIPLEM INFOTECH stands out with its advanced analytics solutions and training programs. The company leverages sophisticated algorithms and machine learning techniques to transform raw data into actionable insights, empowering businesses to make data-driven decisions. Their Data Science training programs are designed to equip individuals with expertise in statistical analysis, predictive modeling, and big data technologies, ensuring they are well-prepared to tackle complex data challenges.

**2.4 Contact Information:**

* **ADDRESS :** 5/3 ,6th Avenue, 32nd street Ashok Nagar ,Chennai -600083
* **Email :** [tripleminfotech@gmail.com](file:///C:\\Users\\Admin\\Downloads\\tripleminfotech@gmail.com)
* **Website :** [contact@tripleminfotech.com](mailto:contact@tripleminfotech.com)
* **Phone number :** +91 98846480

**3. COMPANY DOMAIN**

**3.1 Web Development and Design:**

Web Development and Design involve creating and maintaining websites tailored to business needs. This domain encompasses both the aesthetic aspect of websites (Web Design) and the functional aspect (Web Development). Services include building static and dynamic websites, designing landing pages, and ensuring websites are interactive and responsive. By leveraging modern technologies and design principles, web development and design aim to enhance user experience, increase engagement, and drive business growth.

**3.2 Mobile App Development:**

Mobile App Development focuses on creating applications for mobile devices, such as smartphones and tablets. This domain includes the design, development, testing, and deployment of mobile apps across various platforms (iOS, Android). Mobile apps are crafted to meet specific business needs, providing users with a seamless and intuitive interface, enhancing customer engagement, and offering functionality on-the-go. This service is essential for businesses looking to expand their digital presence and reach a broader audience.

**3.3 Digital Marketing:**

Digital Marketing is a multifaceted domain that involves promoting products or services through digital channels. This includes Search Engine Optimization (SEO), Search Engine Marketing (SEM), Social Media Marketing (SMM), SMS and Email Marketing, and comprehensive branding strategies. Digital marketing aims to increase online visibility, drive traffic to websites, and convert visitors into clients. It’s particularly beneficial for startups and businesses looking to establish a strong online presence and effectively reach their target audience.

**3.4 E-Commerce Solutions:**

E-Commerce Solutions encompass the creation and integration of online shopping platforms, enabling businesses to sell products and services over the internet. This domain includes building ecommerce websites, integrating payment gateways (such as Pay-U-Money, PayPal), and connecting with major e-commerce platforms (Amazon, Flipkart). E-commerce solutions help businesses transition from offline to online sales, expanding their market reach and providing customers with a convenient shopping experience.

**3.5 Web and Email Hosting:**

Web and Email Hosting services provide the infrastructure needed to host websites, web applications, and email systems on the internet. This domain includes managing cloud servers, offering various hosting packages, and ensuring reliable performance and security. Hosting services are crucial for maintaining an online presence, ensuring that websites and emails are accessible and functioning smoothly for customers and businesses alike.

**3.6 On-Demand Online Training:**

On-Demand Online Training involves providing educational and training resources through digital platforms. This domain includes an e-learning portal offering courses and training programs for global students and corporate teams. It supports online training with features such as minimal cost registration for instructors and interactive learning environments. This service caters to the growing need for flexible and accessible education and professional development.

**3.7 Innovative Technologies/IoT:**

Innovative Technologies and the Internet of Things (IoT) involve developing and implementing advanced technological solutions that connect and automate devices and systems. This domain includes research and development in IoT, creating devices that communicate and interact over the internet to enhance efficiency and innovation in various applications. These technologies are pivotal in transforming industries through smarter, data-driven solutions.

**3.8 Branding and Promotional Materials:**

Branding and Promotional Materials focus on creating visual and physical assets that represent and promote a business. This domain includes designing and printing business cards, letter pads, envelope covers, banners, and pamphlets. Effective branding materials are essential for establishing a professional image, engaging customers, and supporting marketing efforts.

**3.9 Electronics and IoT Products:**

Electronics and IoT Products encompass the development and sale of electronic components and modules used in various applications. This domain includes products like GPS modules, Raspberry Pi kits, motion sensors, temperature and humidity sensors, GSM modules, and soil moisture sensors. These products support innovations in IoT, robotics, and other technological fields, providing the building blocks for developing advanced systems and solutions.

**4. USAGE OF COMPUTER:**

Flask has a simple and intuitive design that is easy to understand even if you’re new to it. The flexibility and customization options make it an excellent choice for building small to medium-sized applications that require rapid development and deployment. It supports various HTTP request protocols and has a built-in development server that facilitates the testing and debugging of applications. It also integrates seamlessly with third-party libraries and extensions, providing developers with access to a vast ecosystem of tools and resources that they can leverage to build robust and scalable applications. It allows developers to focus on building their applications' core functionalities without worrying about the underlying infrastructure. This makes it an excellent choice for projects with tight deadlines and limited resources.

**4.1 Development Environments:**

* **PyCharm:** I primarily used PyCharm, a powerful Integrated Development Environment (IDE) for Python, to develop, test, and debug my projects. PyCharm's advanced features, such as code completion, project navigation, and integrated version control, greatly improved my coding efficiency and accuracy.
* **IDLE:** Additionally, I utilized IDLE, Python's Integrated Development and Learning Environment, for quick scripting and testing of Python code. IDLE's simplicity and ease of use made it ideal for running small scripts and testing individual code snippets.

**4.2 Python Web Framework:**

* **Flask:** I used Flask, a lightweight and flexible web framework for Python, to develop web applications. Flask's modular design and ease of integration with other libraries enabled me to build robust web interfaces and deploy machine learning models effectively.

**4.3 Graphical User Interface (GUI) Development:**

* **Tkinter**: For creating graphical user interfaces, I employed Tkinter, Python’s standard GUI toolkit. Tkinter's straightforward approach to building GUI applications allowed me to design and implement interactive and user-friendly interfaces for various projects.

**4.4 API Integration and Data Handling**:

* **Data Processing Libraries:** I leveraged data processing libraries such as pandas for handling and pre-processing data, ensuring the seamless integration of data into my applications.

**4.5 Coding environment:**

The combination of advanced development environments, powerful frameworks, and efficient data handling tools enabled me to successfully complete my projects at TRIPLEM INFOTECH. The use of PyCharm, IDLE, Flask, Tkinter, and Git, along with the supportive operating systems, provided a comprehensive and efficient platform for software development and learning. PyCharm, developed by Jet Brains, is a robust Integrated Development Environment (IDE) specifically designed for Python programming. It provides a comprehensive suite of tools that cater to the needs of both beginners and professional developers. PyCharm enhances productivity through features such as intelligent code completion, which predicts and suggests code snippets, and advanced code navigation, allowing developers to quickly find and jump to different parts of their codebase. The IDE includes an integrated debugger for efficient error detection and resolution, and it supports various version control systems like Git, enabling seamless collaboration and code management. PyCharm’s support for web development frameworks such as Django and Flask, along with its built-in tools for database management, make it a versatile choice for web developers. Additionally, it offers powerful refactoring tools, automated code inspections, and unit testing capabilities, ensuring code quality and maintainability. The ability to customize the environment with plugins and themes further tailors the IDE to individual workflows, making PyCharm an indispensable tool for Python development.

**4.6 Key Features of PyCharm:**

* **Code Editor:** PyCharm offers advanced syntax highlighting for Python and other languages, making the code more readable and easier to debug.
* **Code Completion:** The IDE provides intelligent code completion, suggesting possible completions based on the context, reducing typing effort and errors.
* **Virtual Environments:** PyCharm integrates seamlessly with virtual environments, helping you manage dependencies and package versions for different projects.
* **Step Execution:** The debugger allows you to step through your code line by line or function by function to diagnose issues.
* **Git Support:** PyCharm has built-in support for Git, enabling you to commit, push, pull, merge, and handle branches directly within the IDE.
* **Database Integration:** PyCharm provides tools for managing databases directly from the IDE, including running queries and browsing database tables.
* **Framework Support:** PyCharm supports web frameworks such as Django, Flask, and Pyramid. It provides specialized tools and templates for web development.
* **HTML, CSS, JavaScript:** In addition to Python, PyCharm offers robust support for HTML, CSS, JavaScript, and other web technologies.

**5. ROLE IN THE COMPANY:**

During my internship at TRIPLEM INFOTECH, I successfully completed four projects, actively contributing to the company's initiatives in the field of machine learning and Python programming. My role involved:

* **Utilizing Python Libraries**: Leveraged various Python libraries such as Flask and Tkinter to develop applications and tools.
* **Machine Learning Predictions**: Developed and implemented machine learning models to predict website usage patterns and enhance app functionalities.
* **Application Development**: Created multiple applications for predictive analytics and QR code generation, providing practical solutions for the company's needs.
* **Collaborative Development**: Worked closely with the development team to integrate machine learning algorithms into web applications, improving user experience and operational efficiency.

This experience allowed me to hone my skills in Python programming, machine learning, and application development, contributing to the overall success of the projects and the company's technological advancements.

**5.1 Project Overview:**

**5.1.1 WEBSITE PREDICTION USING DIFFERENT DATASETS:**

Through this project, I gained valuable experience in deploying machine learning models in real-world applications and developed a comprehensive understanding of data handling, model training, and integration within a web framework.

**5.1.2 WEATHER APP:**

In this project, I created a weather application that predicts weather conditions for different cities using data from OpenWeather.org The project involved the following key aspects like API Integration, GUI Development, Data Parsing so I gained some practical experience in combining multiple technologies to create a comprehensive application, enhancing my overall programming and development skills.

**5.1.3 QR CODE GENERATOR:**

This project enhanced my understanding of image generation and processing in Python, as well as the practical application of QR codes for various use cases.

**5.1.3 Google Translator:**

Through this project, I gained practical experience in combining multiple technologies to create a comprehensive application, enhancing my overall programming and development skills.

**5.2 Project Details:**

**Project 1: PREDICTION USING DATASETS**

* **OVERVIEW:** This project focuses on predicting website usage patterns by leveraging different datasets and implementing the solution using Flask, a lightweight web framework in Python they provide valuable insights into the practical applications of machine learning and web development, combining them to create a functional and effective predictive tool.
* **Goal:** The primary objective of this project was to develop a predictive tool for website usage, utilizing various datasets and implementing the solution through Flask. The specific goals include Accurate Predictions, **User-Friendly Interface.**
* **Tasks:** This project provide task such asData Collection and Preparation, Model Development, Model Deployment, Testing and Validation, Real-Time Prediction
* **Machine Learning Algorithms:** You can select algorithm for your dataset
* **Visualization:** Data visualization libraries were been used to create insightful charts and visual representations of the market segments.
* **Technical skill :** Gained proficiency in data pre-processing, Python programming, machine learning algorithms and data visualization
* **Outcomes:** Successfully deployed a machine learning model within the web application using Flask to provide accurate usage predictions.

**Project 2: WEATHER PREDICTION APP FOR DIFFERENT CITIES**

* **Overview:** This project involved developing a weather prediction application capable of providing forecasts for different cities. Utilizing data from OpenWeather.org API, the application generates predictions for various weather conditions based on the user's input of city names
* **Goal**: These goals outline the project's objectives and set a clear direction for its development and implementation. Such as Accurate Predictions, **User-Friendly Interface,** Reliable Performance, User Satisfaction.
* Tasks: this project provide task such as API Integration, Cross-Platform Compatibility
* **Technical skills**: Gained proficiency of understanding various aspects of application development, data analysis, and prediction generation in the context of weather forecasting.
* **Accurate Predictions**: They generate accurate weather predictions for different cities.
* **Skill Development**: Enhanced your skills in API integration, data processing, algorithm development, and application deployment.
* **User Satisfaction**: Provided users with a valuable tool for accessing reliable weather forecasts, enhancing their overall satisfaction.
* **Outcomes:** Successfully delivered a fully functional Weather App Prediction applications.

**Project 3: QR CODE GENERATOR**

* **Overview:** QR code generators are versatile tools that offer a convenient way to encode and share information in a compact, scan able format, making them valuable for a wide range of applications across various industries.
* **Data Encoding**: QR code generators encode data into QR codes, allowing users to create codes containing text, URLs, phone numbers, email addresses, Wi-Fi credentials, and other types of information.
* **Customization**: Users can often customize the appearance of QR codes by adjusting parameters such as size, colour, error correction level, and logo placement.
* **High Compatibility**: QR code generators typically support various output formats, including PNG, JPEG, SVG, and PDF, making it easy to use QR codes across different platforms and devices.
* **Goal**: Enable Efficient Information Sharing AND the goal is to facilitate seamless communication and information exchange across different platforms and devices.
* **Tasks: The project provide task that** aim to enhance accessibility and convenience for users by enabling them to create and share QR codes quickly and easily, thereby streamlining processes and improving communication efficiency.
* **Technical skills:** By understanding these technical skills through the development of a QR code generator, you not only enhance your capabilities as a developer but also gain practical experience in building applications that serve real-world needs.
* **Outcomes:** Successfully you can provide a versatile and convenient tool for users to encode and share information effectively.

**Project 4 : Google Translator:**

* **Overview**:This project involved developing a Google Translator application capable of translating text between various languages. Utilizing the Google Translate API, the application provides accurate and real-time translations based on the user's input of text and selection of target languages.
* **Goal**:These goals outline the project's objectives and set a clear direction for its development and implementation. Specific goals included achieving Accurate Translations, User-Friendly Interface, Reliable Performance, and User Satisfaction.
* **Tasks**:This project included tasks such as API Integration, GUI Development, and Multi-Language Support to ensure comprehensive and efficient translation services.
* **Technical Skills**:Gained proficiency in understanding various aspects of application development, data processing, and real-time translation in the context of language translation services.
* **Accurate Translations**: The application delivers precise and reliable translations for various languages
* **Skill Development**: Enhanced skills in API integration, text processing, GUI design, and application deployment.
* **User Satisfaction**: Provided users with a valuable tool for accessing accurate translations, enhancing their overall communication and understanding.
* **Outcomes**: Successfully delivered a fully functional Google Translator application, capable of providing real-time translations for multiple languages.

**6. LEARNING DURING THE INTERNSHIP:**

My internship at TRIPLEM INFOTECH provided a transformative learning experience that enriched my understanding of various aspects within the field of machine learning and Python programming. Throughout my tenure, I acquired valuable insights and skills in the following areas:

**6.1 Flask:**

Flask is a lightweight and versatile Python web framework that simplifies the process of building web applications. It is renowned for its simplicity, flexibility, and ease of use, making it a popular choice among developers for projects ranging from small prototypes to large-scale applications. Flask provides the essential tools needed for web development while allowing developers, the freedom to structure their applications according to their preferences. Flask remains a powerful and versatile choice for web development, offering developers the flexibility to create custom solutions tailored to their specific requirements. With its simplicity, extensibility, and vibrant community, Flask continues to play a significant role in the ever evolving landscape of web development.

**6.2 Flask Framework:**

Flask distinguishes itself as a micro-framework, meaning it prioritizes simplicity and

Minimalism. Unlike more opinionated frameworks like Django, Flask does not impose a

Specific project structure or require the use of certain tools. This flexibility allows developers to choose the components and libraries that best suit their needs, making Flask adaptable to a wide range of use cases.

**6.3 Core Concepts:**

At the heart of Flask are its core concepts:

❖ **Routes and view functions:** URL routing in Flask is achieved through route

Decorators, which map URL patterns to Python functions (view functions) that generate

HTTP responses.

❖ **Request handling and response generation:** Flask provides tools for handling

Incoming HTTP requests and generating appropriate responses.

❖ **Template rendering:** Flask supports template rendering using Jinja2, a powerful and

Flexible templating engine that allows developers to create dynamic HTML content.

❖ **Configuration management:** Flask applications can be configured using simple

Python scripts, allowing developers to customize behavior based on different

Environments (e.g., development, production).

❖ **Error handling:** Flask provides mechanisms for handling errors gracefully, including

Custom error pages and exception handling.

**6.4 MVC Architecture:**

While Flask does not enforce a strict MVC architecture, it is often used in a manner

Consistent with this pattern. In Flask:

❖ **Model:** Data models and database interactions are typically managed using libraries

Like SQLAlchemy.

❖ **View:** View functions in Flask serve as controllers, handling requests and generating

responses.

❖ **Controller:** While Flask doesn't have a dedicated controller layer, the logic for

Processing requests and coordinating interactions between models and views is

encapsulated within view functions.

**6.5 Project Structure:**

Flask projects commonly follow a modular structure, separating concerns such as routes, models, and templates into distinct modules or packages. A typical Flask project might include directories for:

❖ **Routes:** Contains route definitions and view functions.

❖ **Models:** Contains data models and database interactions.

❖ **Templates:** Contains HTML templates for rendering dynamic content.

❖ **Static:** Contains static assets such as CSS, JavaScript, and images.

**6.6 Extensions and Libraries:**

Flask's extensibility is one of its greatest strengths, thanks to a vibrant ecosystem of

Third-party extensions and libraries. Some popular Flask extensions include:

❖ **Flask-SQLAlchemy:** Integrates Flask with SQLAlchemy, a powerful and flexible

ORM for working with databases.

❖ **Flask-WTF:** Provides utilities for handling web forms and performing form validation.

❖ **Flask-Login:** Simplifies user authentication and session management.

❖ **Flask-RESTful:** Facilitates the development of RESTful APIs using Flask

**6.7 Integration with Machine Learning:**

Flask's flexibility and extensibility make it an excellent choice for integrating machine learning models into web applications. By combining Flask with machine learning libraries like TensorFlow, scikit-learn, or PyTorch, developers can create powerful applications that leverage predictive analytics, natural language processing (NLP), computer vision, and more.

❖ **Model Deployment:** Flask provides a convenient platform for deploying machine

learning models as RESTful APIs. Developers can use Flask to create endpoints that

receive input data, pass it through a pre-trained machine learning model, and return the

model's predictions or classifications.

❖ **Real-time Predictions:** With Flask-SocketIO or similar libraries, developers can

enable real-time communication between web clients and machine learning models.

This allows for dynamic updates and interactive experiences based on model

predictions.

❖ **Data Visualization:** Flask's integration with libraries like Plotly or Bokeh enables

developers to create interactive data visualizations that showcase the results of machine

learning analyses. These visualizations can be embedded directly into Flask-powered

web applications.

❖ **Personalization and Recommendation Systems:** Flask applications can leverage

machine learning algorithms to provide personalized recommendations to users based

on their behavior, preferences, or historical data. These recommendation systems can

be integrated seamlessly into Flask-powered web applications to enhance user

engagement.

❖ **Natural Language Processing (NLP) and Sentiment Analysis:** Flask can be used to

build web applications that perform text analysis tasks such as sentiment analysis, topic

modeling, or named entity recognition. By integrating Flask with NLP libraries like NLTK or spacy, developers can create web-based tools for analysing and visualizing

textual data.

**6.8 GUI Development**:

I developed a user-friendly graphical user interface (GUI) using Tkinter. This experience taught me how to design and implement intuitive interfaces that allow users to easily interact with the application.

* **Data Handling and Processing:** I gained experience in handling and processing JSON data received from the API. This included parsing the JSON data to extract relevant weather information and displaying it in a meaningful way within the application.
* **Python Programming:** The project enhanced my Python programming skills, particularly in working with external APIs and developing GUIs. I became more proficient in writing clean, efficient, and maintainable code.
* **GUI Design**: I learned the principles of designing and implementing an intuitive and functional GUI using Tkinter. This involved creating widgets, handling user inputs, and updating the interface dynamically based on the data received.
* **Third-Party API Usage**: Learned how to integrate and utilize third-party APIs, such as the Google Charts API, to enhance the functionality of applications.
* **Data Encoding and Decoding:** Gained knowledge about encoding different types of data into QR codes, such as URLs, text, and contact information. Learned various techniques for encoding data effectively into QR code format.
* **Customization**: Acquired skills in customizing QR codes, including adjusting size, color, error correction levels, and adding logos

# 6.11 CODE IMPLEMENTATION:

PROJECT-1:

TITLE: WEBSITE PREDICTION USING FLASK

CODE :

App.py:

**IMPORT LIBRARIES :**

import numpy as np

from flask import \*

from sklearn.metrics import mean\_squared\_error, r2\_score

import pandas as pd

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestRegressor

from sklearn.model\_selection import train\_test\_split

import pickle

import warnings

import seaborn as sns

import matplotlib.pyplot as plt

import io

import base64

from flask\_sqlalchemy import SQLAlchemy

import csv

**# Load the dataset**

df = pd.read\_csv("Student\_Marks2.csv")

**# Separate features (X) and target variable (y)**

X = df[['number\_courses', 'time\_study']]

y = df['Marks']

**# Split the data into training and testing sets**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=50)

**# Standardize features**

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

**# Train the Random Forest regressor**

regressor = RandomForestRegressor()

regressor.fit(X\_train, y\_train)

**# Save the trained model to a file**

pickle.dump(regressor, open("student\_marks\_model.pkl", "wb"))

# **Flask Application Initialization**

app = Flask(\_\_name\_\_,static\_url\_path='/static')

app.config["SQLALCHEMY\_DATABASE\_URI"] = "sqlite:///project.db"

db = SQLAlchemy(app)

**#SQLAlchemy Model Definition**

class User3(db.Model):

id = db.Column(db.Integer, primary\_key=True)

username = db.Column(db.String(30),nullable=True)

password = db.Column(db.String(30),nullable=True)

Email = db.Column(db.String(30),nullable=True)

**#Define signup route**

@app.route('/', methods=['POST', 'GET'])

def signup():

if request.method == 'POST':

username = request.form['name'] #Get username from the form

password = request.form['pass']#Get password from the form

Email = request.form['email']

**#create user3 model**

user = User3(username=username, password=password, Email=Email)

**#create a table**

db.create\_all()

**#create new user**

db.session.add(user)

**#save the new user**

db.session.commit()

return redirect(url\_for('login'))

return render\_template("signup.html")

**#define login route**

@app.route('/login', methods=['POST', 'GET'])

def login():

if request.method == 'POST':

Email = request.form['email']

password = request.form['pass']

user = User3.query.filter\_by(Email=Email).first()

if user and user.password == password:

return redirect(url\_for('home'))

else:

return "Login Failed!"

return render\_template("login.html")

**#define home route**

@app.route('/home')

def home():

return render\_template('home.html')

**#deploy model render page**

@app.route('/model', methods=['GET', 'POST'])

def model():

if request.method == 'POST' and 'deploy\_model' in request.form:

return redirect(url\_for('predict'))  **# Redirect to the predict page**

return render\_template('model.html')

**#Describe model page**

@app.route('/model2.html')

def model2():

return render\_template('model2.html')

**#define contact page**

@app.route('/contact.html')

def contact():

return render\_template('contact.html')

**#define about page**

@app.route('/about.html', methods=['GET', 'POST'])

def about():

if request.method == 'POST' and 'dataset\_name' in request.form:

**# Redirect to read endpoint with an identifier indicating data should be displayed**

return redirect(url\_for('read', show\_data=True))

return render\_template('about.html')

**#reading the csv file from the dataset**

@app.route('/read')

def read():

show\_data = request.args.get('show\_data', False)

if show\_data:

data = []

# Read data from CSV file

with open("Student\_Marks2.csv", newline='') as csvfile:

reader = csv.DictReader(csvfile)

for row in reader:

data.append(row)

# Pass data to the template

return render\_template('about.html', data=data)

else:

# If show\_data parameter is not present, redirect back to about page

return redirect(url\_for('about'))

**#prediction from the dataset:**

@app.route('/predict', methods=["GET", "POST"])

def predict():

if request.method == 'POST':

# Extract features from the form

number\_courses = float(request.form['number\_courses'])

time\_study = float(request.form['time\_study'])

**# Load the trained model**

model = pickle.load(open("student\_marks\_model.pkl", "rb"))

**# Make prediction**

prediction = model.predict([[number\_courses, time\_study]])[0]

**# Calculate evaluation metrics using the provided data and the predicted values**

y\_pred = model.predict(X)

mse = mean\_squared\_error(y, y\_pred)

r2 = r2\_score(y, y\_pred)

**#generate plotting**

pairplot = generate\_pairplot(df)

heatmap = generate\_heatmap(df)

boxplot = generate\_boxplot(df)

bar\_chart = generate\_bar\_chart(df)

pie\_chart = generate\_pie\_chart(df)

return render\_template('predict.html',

prediction\_text="Predicted marks: {:.2f}".format(prediction),

mse\_text="Mean Squared Error: {:.2f}".format(mse),

r2\_text="R-squared: {:.2f}".format(r2),

pairplot=pairplot, heatmap=heatmap,

boxplot=boxplot, bar\_chart=bar\_chart, pie\_chart=pie\_chart)

return render\_template('predict.html')

**#generate pairplot visualization**

def generate\_pairplot(df):

sns.pairplot(df)

buf = io.BytesIO()

plt.savefig(buf, format='png')

buf.seek(0)

pairplot = base64.b64encode(buf.read()).decode('utf8')

return pairplot

**#generate heatmap visualization**

def generate\_heatmap(df):

plt.figure(figsize=(8, 6))

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

buf = io.BytesIO()

plt.savefig(buf, format='png')

buf.seek(0)

heatmap = base64.b64encode(buf.read()).decode('utf8')

return heatmap

**#generate boxplot visualization**

def generate\_boxplot(df):

plt.figure(figsize=(8, 6))

sns.boxplot(data=df)

buf = io.BytesIO()

plt.savefig(buf, format='png')

buf.seek(0)

boxplot = base64.b64encode(buf.read()).decode('utf8')

return boxplot

**#generate bar chart picture visualization**

def generate\_bar\_chart(df):

plt.figure(figsize=(8, 6))

sns.barplot(data=df)

buf = io.BytesIO()

plt.savefig(buf, format='png')

buf.seek(0)

bar\_chart = base64.b64encode(buf.read()).decode('utf8')

return bar\_chart

**#generate pie chart visualization**

def generate\_pie\_chart(df):

plt.figure(figsize=(8, 6))

df['Marks'].value\_counts().plot.pie(autopct='%1.1f%%')

buf = io.BytesIO()

plt.savefig(buf, format='png')

buf.seek(0)

pie\_chart = base64.b64encode(buf.read()).decode('utf8')

return pie\_chart

**# main function for the flask to run**

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

**Predict.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Prediction</title>

<link rel ="stylesheet" href="{{url\_for('static',filename='css/model.css')}}">

</head>

<body>

<section class="three">

<h1>Student Marks Prediction</h1>

<form action="{{ url\_for('predict') }}" method="post">

<input type="text" name="number\_courses" placeholder="Number of Courses" required>

<input type="text" name="time\_study" placeholder="Time Spent on Study" required>

<button type="submit" value="submit">Predict</button>

</form>

<br>

<br>

{% if prediction\_text %}

<p>{{ prediction\_text }}</p>

{% endif %}

<br>

<br>

<br>

{% if mse\_text %}

<p>MSE: {{ mse\_text }}</p>

{% endif %}

<br>

{% if r2\_text %}

<p>R-squared: {{ r2\_text }}</p>

{% endif %}

<br>

<br>

{% if pairplot %}

<h2>Pairplot</h2>

<img src="data:image/png;base64,{{ pairplot }}" alt="Pairplot">

{% endif %}

<br>

{% if heatmap %}

<h2>Heatmap</h2>

<img src="data:image/png;base64,{{ heatmap }}" alt="Heatmap">

{% endif %}

<br>

{% if boxplot %}

<h2>Boxplot</h2>

<img src="data:image/png;base64,{{ boxplot }}" alt="Boxplot">

{% endif %}

<br>

{% if bar\_chart %}

<h2>Bar Chart</h2>

<img src="data:image/png;base64,{{ bar\_chart }}" alt="Bar Chart">

{% endif %}

<br>

{% if pie\_chart %}

<h2>Pie Chart</h2>

<img src="data:image/png;base64,{{ pie\_chart }}" alt="Pie Chart">

{% endif %}

<br>

</section>

</body>

</html>

**Model2.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>model2 </title>

</head>

<body>

<h1> ABOUT OUR MODEL DEPLOYMENT</h1>

<img src="{{ url\_for('static', filename='pictures/random.png') }}"width="100%" height="100%">

</body>

</html>

**Contact.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>contact</title>

</head>

<body>

<h1> how to contact us </h1>

<h1> <b>contact information</b></h1>

<p> phone num:876543290</p>

<p> <b>Email:</b>modelgmail@123.com</p>

</body>

</html>

**About.html:**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>About Page</title>

<link rel ="stylesheet" href="{{url\_for('static',filename='css/model.css')}}">

</head>

<body>

<div class="container">

<h1>About Page</h1>

<p>This is the about page of our website.</p>

<img src="{{ url\_for('static', filename='pictures/flask.jpg') }}"width="100%" height="100%">

<!-- Form to trigger the dataset viewing -->

<form action="{{ url\_for('about') }}" method="POST">

<!-- Add a hidden input field to indicate the action -->

<input type="hidden" name="dataset\_name" value="Student\_Marks">

<!-- Button to submit the form -->

<button type="submit">View Dataset</button>

</form>

<!-- Display data if it's available -->

{% if data %}

<h2>Dataset:</h2>

<table border="2">

<thead>

<tr>

<th>Number of Courses</th>

<th>Time of Study</th>

<th>Marks</th>

</tr>

</thead>

<tbody>

{% for row in data %}

<tr>

<td>{{ row.number\_courses}}</td>

<td>{{ row.time\_study }}</td>

<td>{{ row.Marks }}</td>

</tr>

{% endfor %}

</tbody>

</table>

{% endif %}

</div>

</body>

</html>

**#home.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Home Page</title>

<link rel ="stylesheet" href="{{url\_for('static',filename='css/model.css')}}">

</head>

<body>

<section class="one">

<ul>

<li><a href="/home">HOME1</a></li>

<li><a href="/model2.html">MODEL</a></li>

<li><a href="/contact.html">CONTACT</a></li>

<li><a href="/about.html">ABOUT</a></li>

</ul>

<h4>LOGIN SOON FOR SALES OFFER 50%</h4>

<img src="{{ url\_for('static', filename='pictures/model.jpg') }}"width="100%" height="100%">

<form action="{{ url\_for('model') }}" method="post">

<input type="submit" name="deploy\_model" value="Deploy Model">

</form>

</section>

</body>

</html>

**#signup.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>sign</title>

<link rel ="stylesheet" href="{{url\_for('static',filename='css/model.css')}}">

</head>

<body>

<h2>User Registration</h2>

<form action="{{url\_for('signup')}}" method="post" >

<label for="name">username</label>

<input type="text" id="name" name="name" required><br><br>

<label for="pass">password</label>

<input type="text" id="pass" name="pass" required><br><br>

<label for="email">Email:</label>

<input type="text" id="email" name="email" required><br><br>

<input type="submit" value="Submit">

</form>

</body>

</html>

**#login.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>login</title>

<link rel ="stylesheet" href="{{url\_for('static',filename='css/model.css')}}">

</head>

<body>

<h1>LOGIN PAGE</h1>

<section class="login">

<form action="{{url\_for('login')}}" method="post">

<label for="pass">password</label>

<input type="text" id="pass" name="pass" required><br><br>

<label for="email">Email:</label>

<input type="text" id="email" name="email" required><br><br>

<input type="submit" value="Submit">

</form>

</section>

</body>

</html>

**Model.css**

ul {

list-style-type: none;

margin: 0;

padding: 0;

overflow: hidden;

background-color: pink;

text-align: center;

border: 10px dashed violet;

display: flex;

}

li {

display: inline;

margin-left: 60px;

}

a{

display: block;

padding: 10px;

}

form{

text-align:center;

background-color:lightblue;

border:20px solid pink;

}

input{

border-radius:50px 60px;

border:2px solid pink;

padding:10px;

}

label{

font-size:20px;

color:brown;

}

h2{

font-size:50px;

font-style:italic;

text-shadow: 2px 2px 5px red, 0 0 2px blue;

text-align:center;

color:lightblue;

border:20px dashed pink;

}

a{

background-color:lightblue;

border : 10px dashed pink;

color: black

padding:20px;

}

.login{

background-color:pink;

padding-bottom:800px;

}

h1{

font-size:50px;

font-style:italic;

text-shadow:2PX 2PX 5PX red,0 0 2px blue;

text-align:center;

color:pink;

border:20px dashed lightblue;

}

button{

color:pink

background-color: lightblue;

border:2px dashed violet;

}

table,th,td{

padding:30px;

width: 10px;

height: 10px;

display: block-inline;

border: 10px solid violet;

color:blue;

margin-left:60px;

}

th{

background-color : pink;

}

.one{

background-color:

}

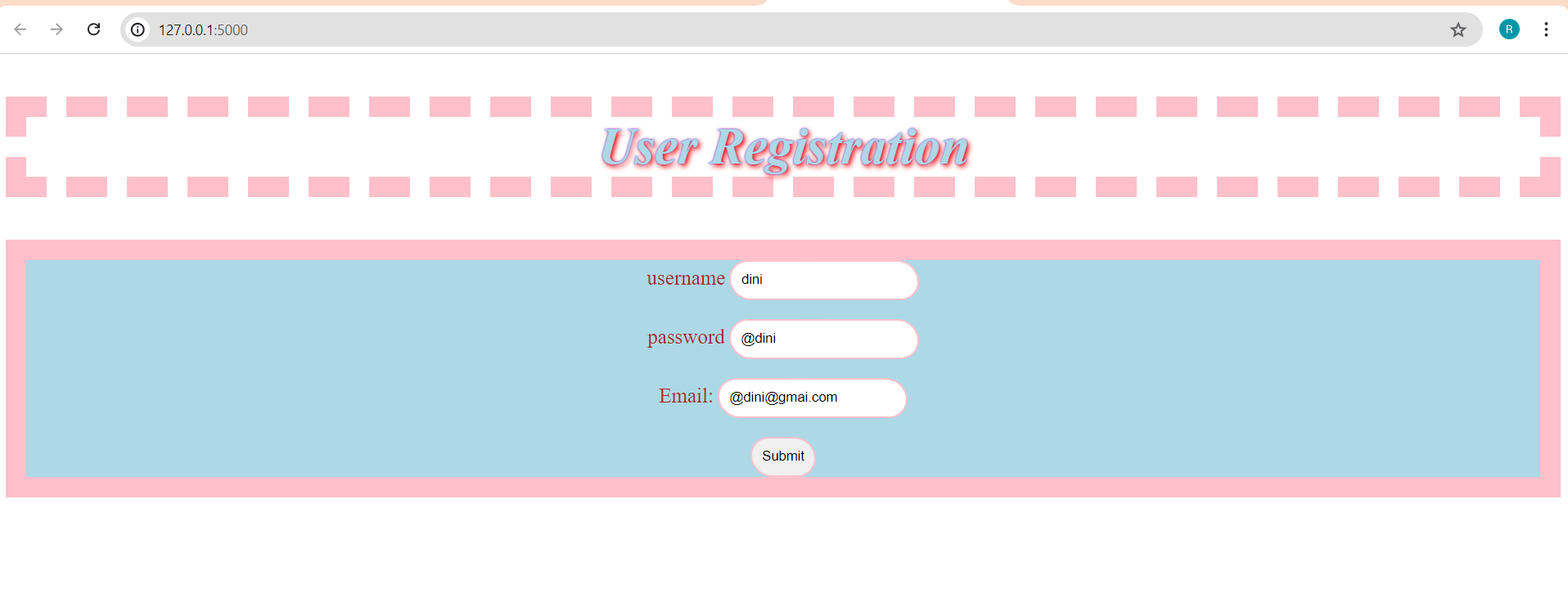
.container{

background-color: #f5f5f5;

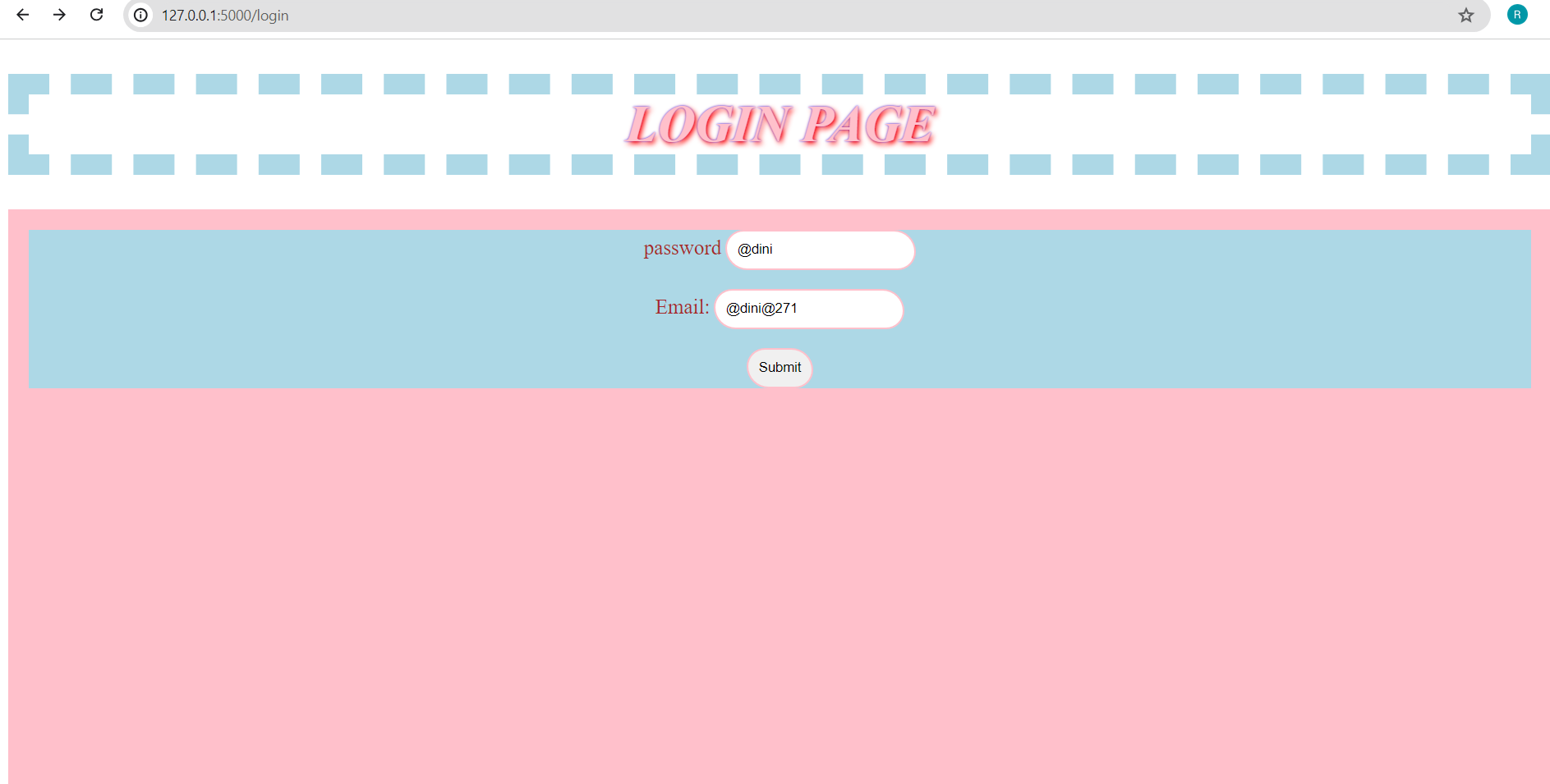
}

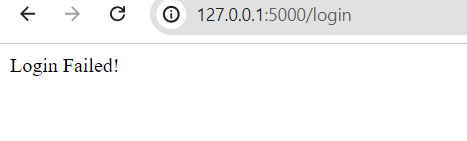
**Output**

**SIGN UP PAGE**

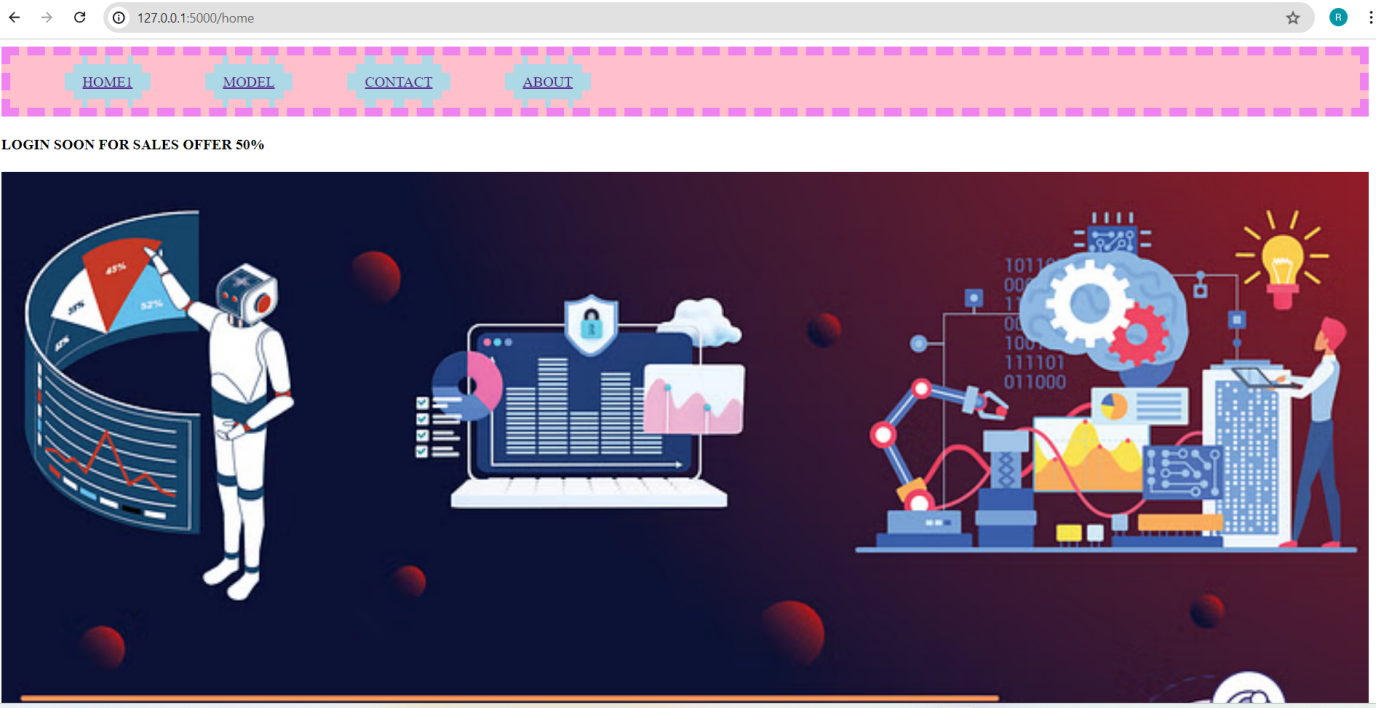


LOGIN PAGE





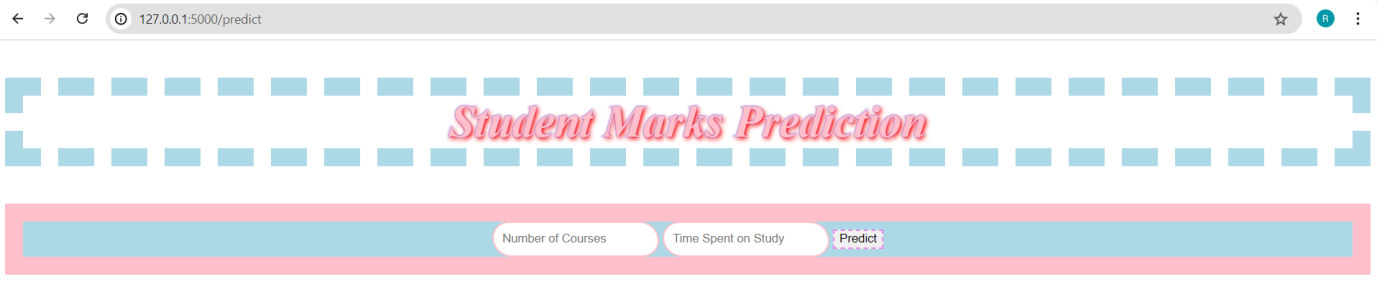
HOME PAGE:



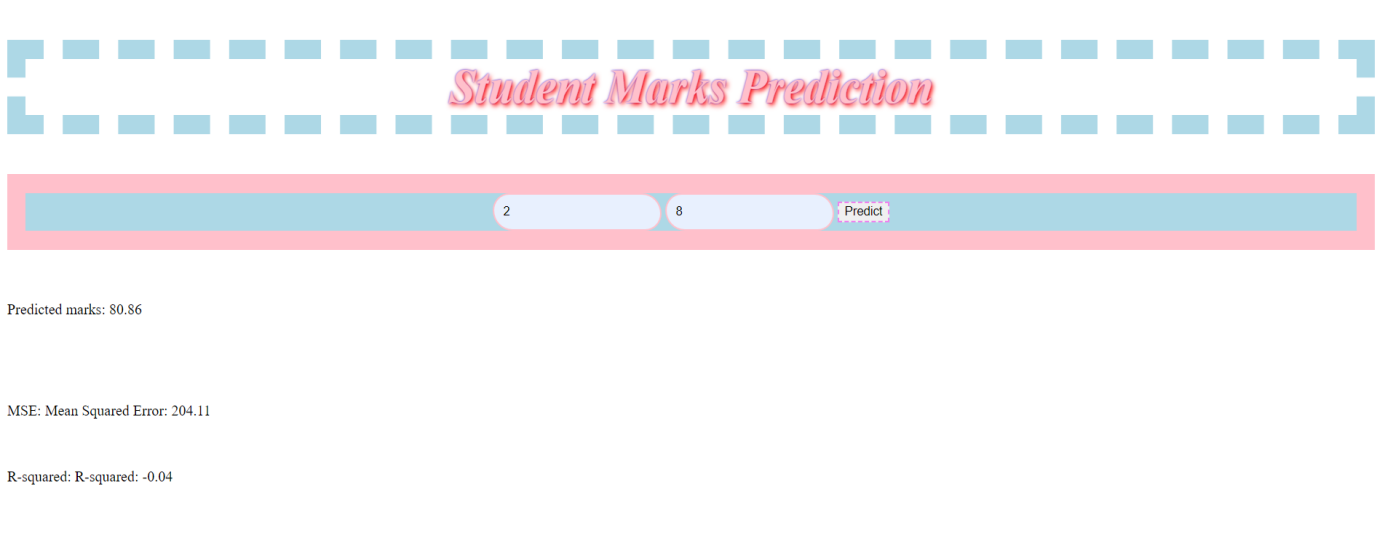


DEPLOY MACHINE LEARNING MODEL

MARKS PREDICTION

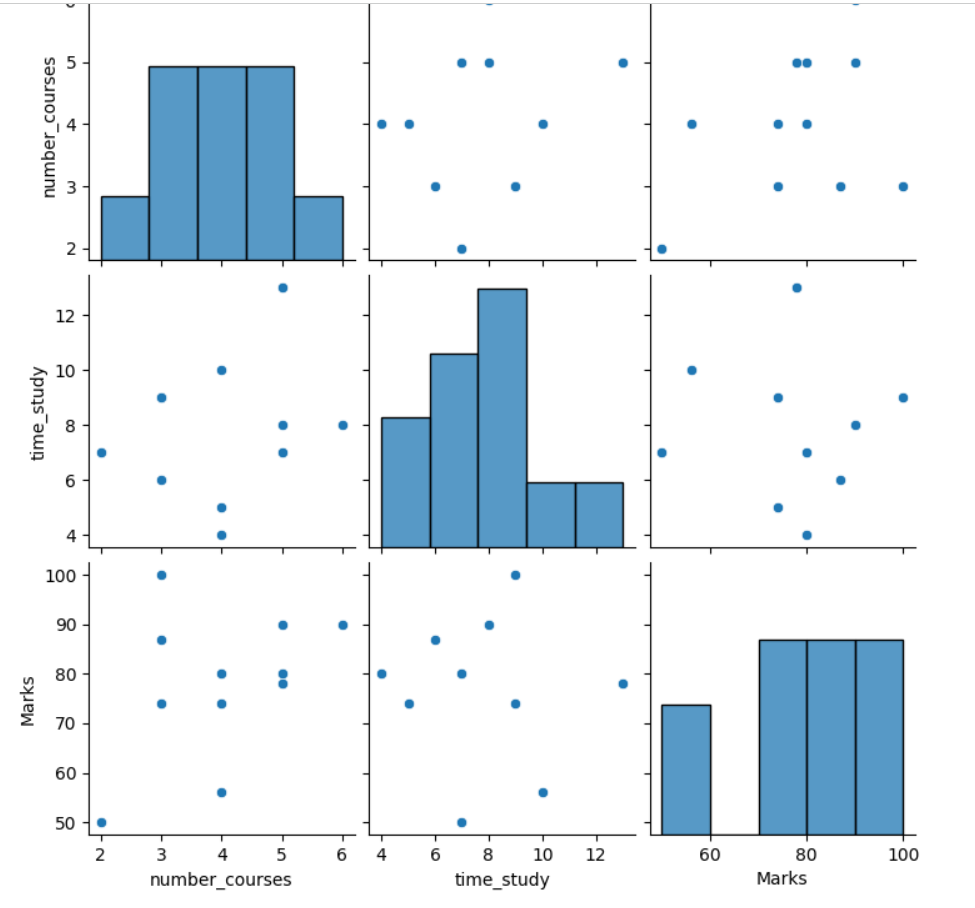


PREDICTION

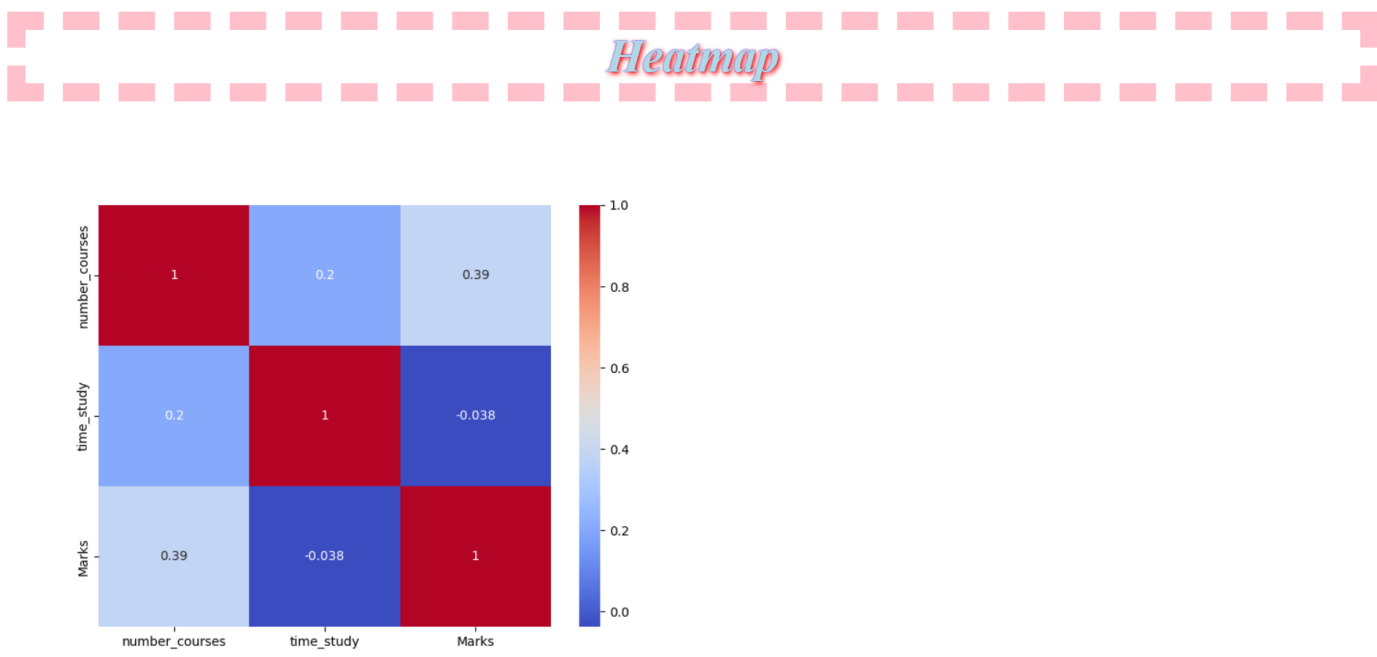


VISUALIZATION

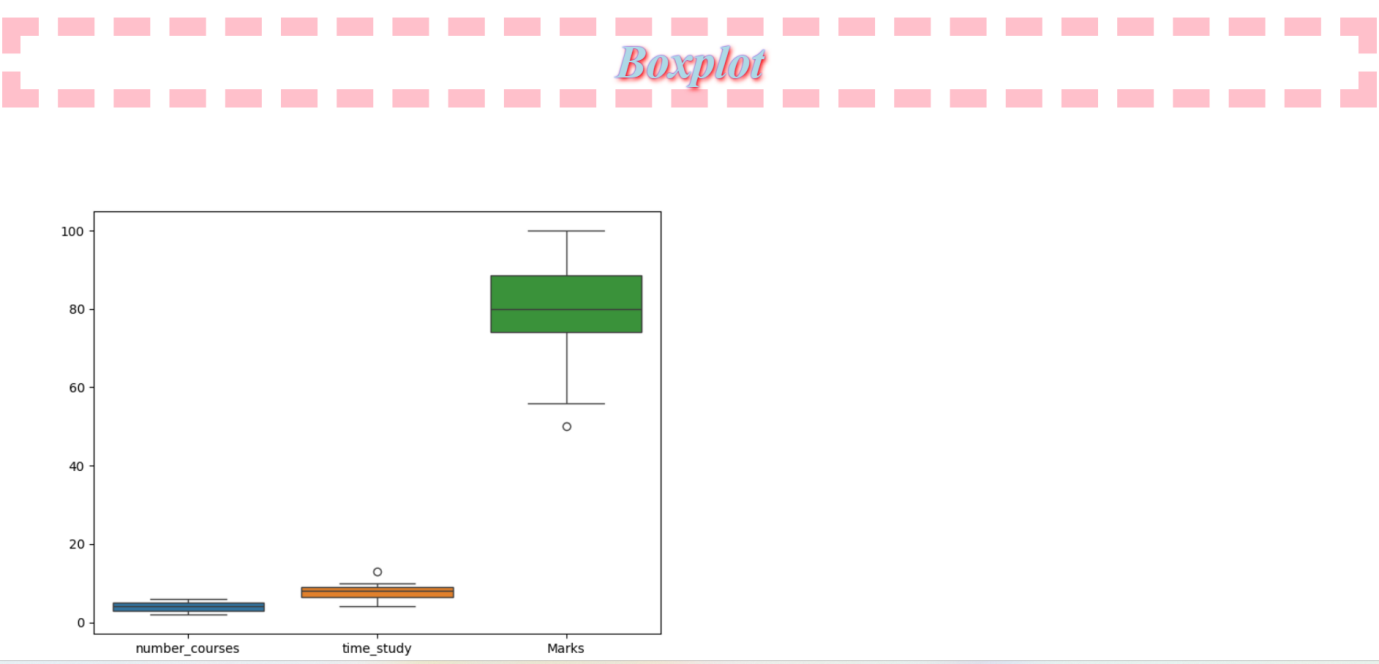
PAIRPLOT:

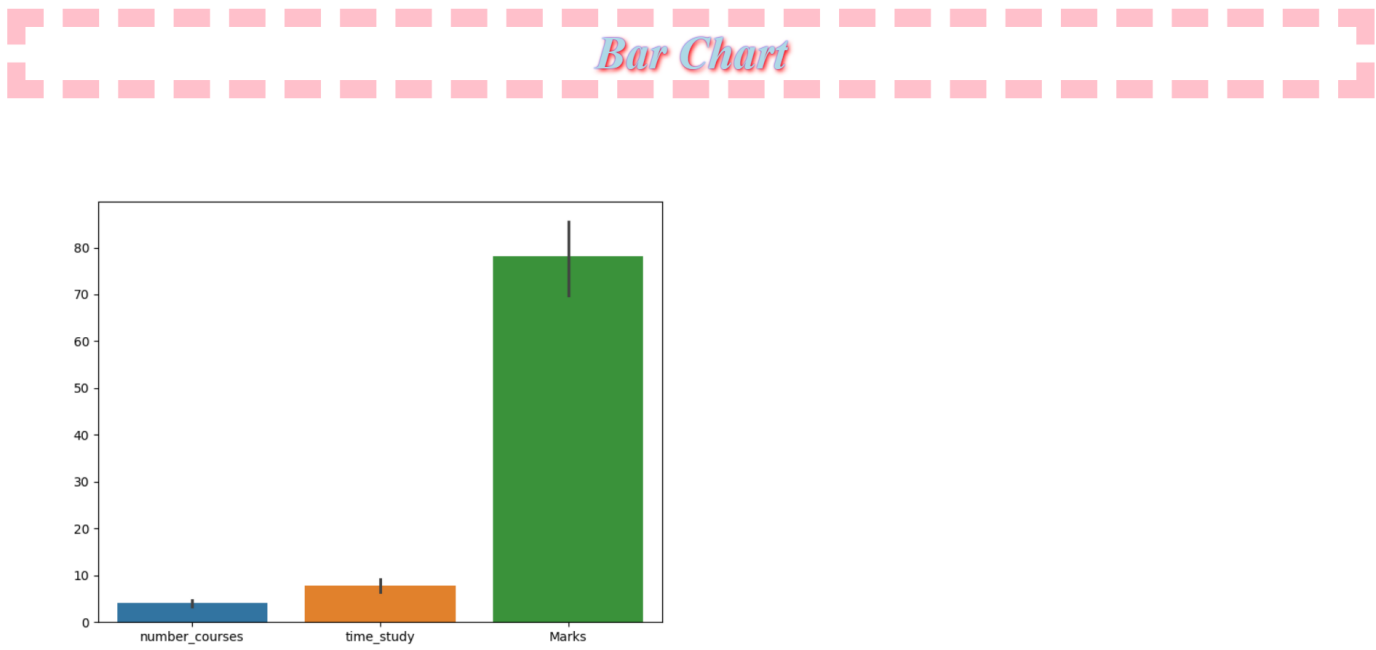


HEATMAP:

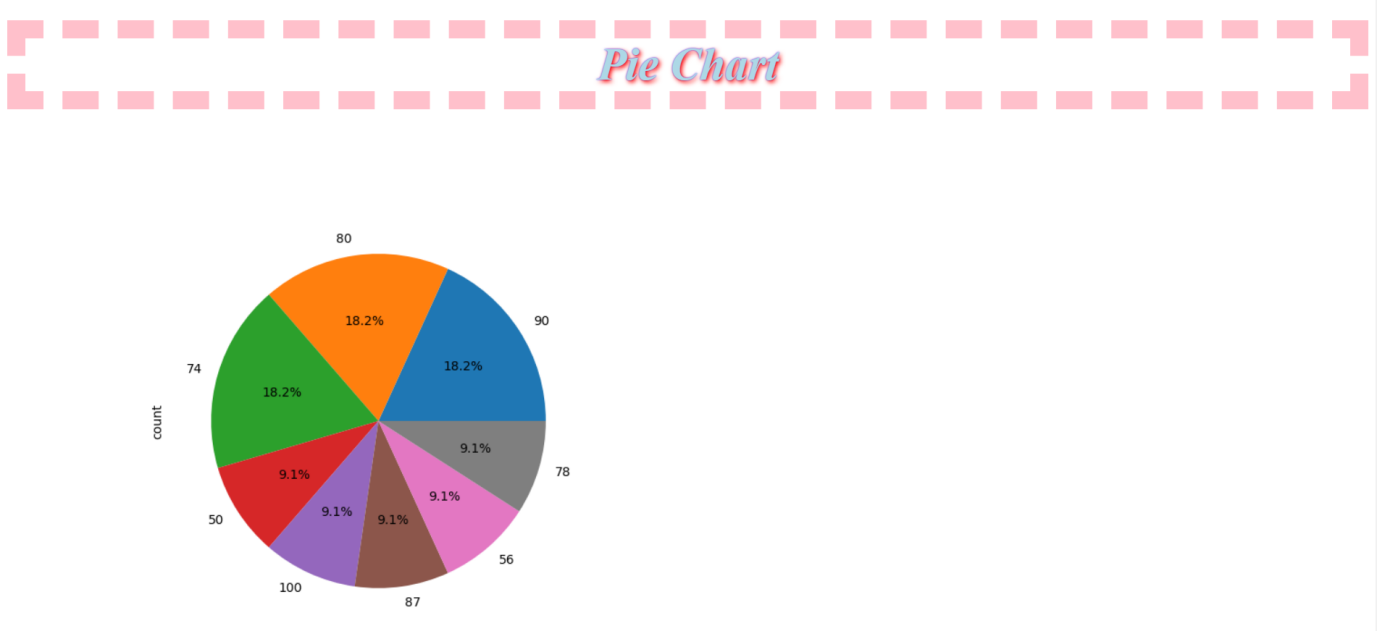


BOXPLOT :

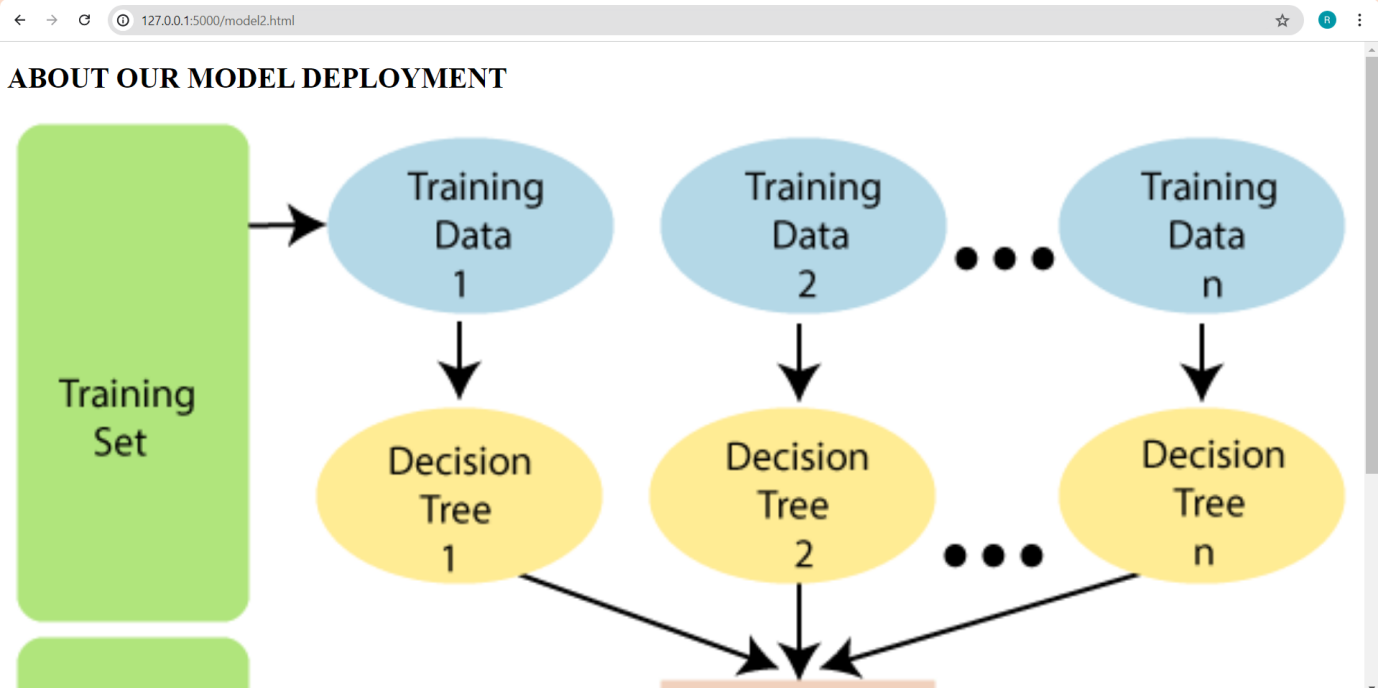


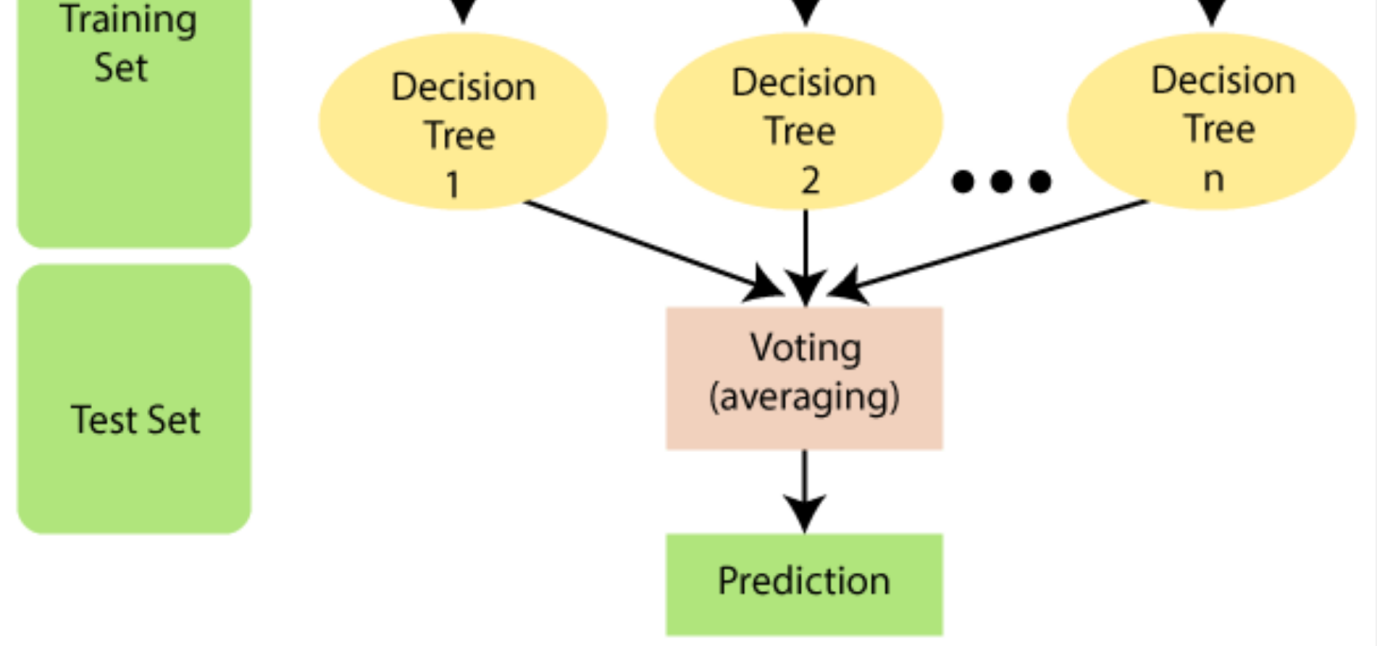
BARCHART

PIE CHART:

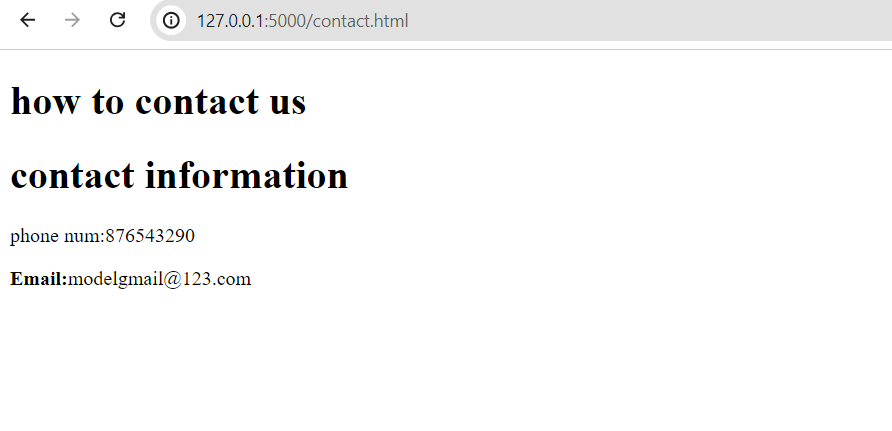


MODEL2 PAGE



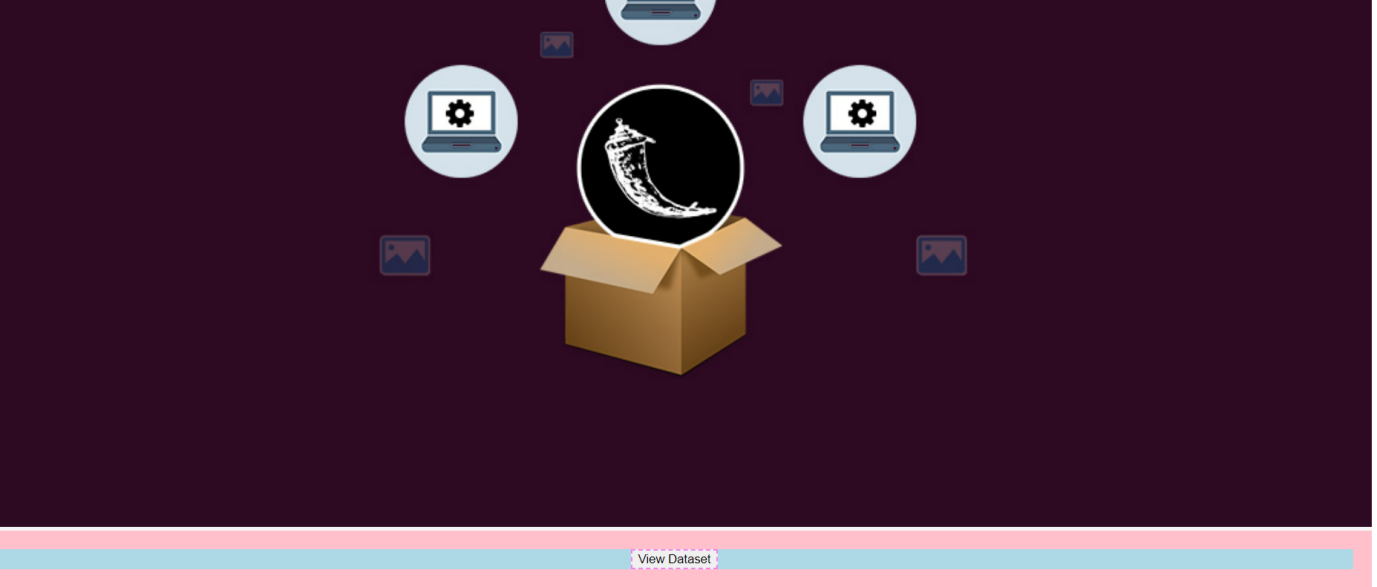


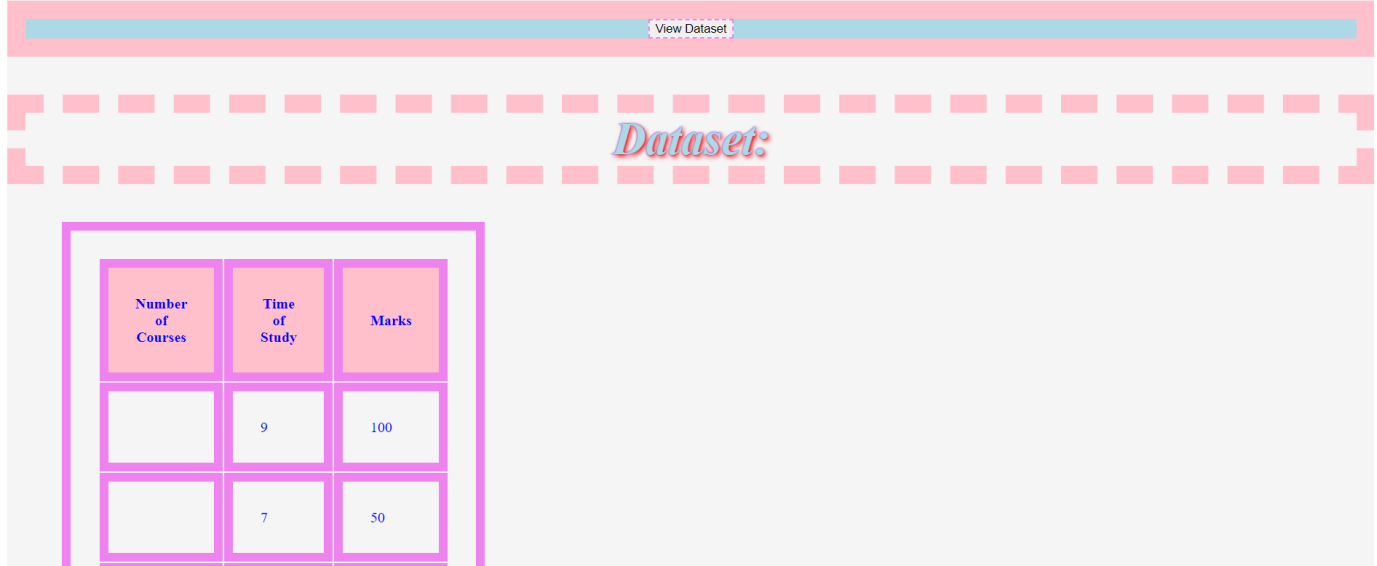
CONTACT PAGE:



ABOUT PAGE:







PROJECT-2:

TITLE: WEATHER APP

CODE :

Demo.py:

from tkinter import \*

from tkinter import ttk

import requests

def data\_get():

city= city\_name.get()

data requests.get("https://api.openweathermap.org/data/2.5/weather?q="+city+"&appid=20a01944eda34a13f4a4dcecfff77197").json() #convert to json file

print(data)

w\_label1.config(text=data["weather"][0]["main"])

wb\_label1.config(text=data["weather"][0]["description"])

temp\_label1.config(text=str(data["main"]["temp"]-273.15))

pressure\_label1.config(text=data["main"]["pressure"])

win = Tk()

win.title(" JESUS TELL US THE WEATHER ")

win.config(bg = "blue")

win.geometry("500x570")

name\_label = Label(win, text="Weather app", font=("Times New Roman", 30, "bold"))

name\_label.place(x=25, y=50, height=50, width=450)

city\_name= StringVar()

list\_names = ["Andhra Pradesh","Arunachal Pradesh ","Assam","Bihar","Chhattisgarh","Goa","Gujarat","Haryana","Himachal Pradesh","Jammu and Kashmir","Jharkhand","Karnataka","Kerala","Madhya Pradesh","Maharashtra","Manipur","Meghalaya","Mizoram","Nagaland","Odisha","Punjab","Rajasthan","Sikkim","Tamil Nadu","Telangana","Tripura","Uttar Pradesh","Uttarakhand","West Bengal","Andaman and Nicobar Islands","Chandigarh","Dadra and Nagar Haveli","Daman and Diu","Lakshadweep","National Capital Territory of Delhi","Puducherry"]

com = ttk.Combobox(win, text="Weather app",values=list\_names , font=("Times New Roman", 20, "bold"),textvariable=city\_name)

com.place(x=25, y=120, height=50, width=450)

w\_label = Label(win,text="Weather climate",font=("Times New Roman", 20, "bold"))

w\_label.place(x=25,y=260, height=50,width=210)

w\_label1 = Label(win,text="",font=("Times New Roman", 20, "bold"))

w\_label1.place(x=250,y=260, height=50,width=210)

wb\_label = Label(win,text="Weather description",font=("Times New Roman", 17, "bold"))

wb\_label.place(x=25,y=330, height=50,width=210)

wb\_label1= Label(win,text="t",font=("Times New Roman", 17, "bold"))

wb\_label1.place(x=25,y=330, height=50,width=210)

wb\_label = Label(win,text="Weather description",font=("Times New Roman", 17, "bold"))

wb\_label.place(x=25,y=330, height=50,width=210)

wb\_label1 = Label(win,text="",font=("Times New Roman", 17, "bold"))

wb\_label1.place(x=250,y=330, height=50,width=210)

temp\_label = Label(win,text="Temperature",font=("Times New Roman", 20, "bold"))

temp\_label.place(x=25,y=400, height=50,width=210)

temp\_label1 = Label(win,text="",font=("Times New Roman", 20, "bold"))

temp\_label1.place(x=250,y=400, height=50,width=210)

pressure\_label = Label(win,text="Presure ",font=("Times New Roman", 17, "bold"))

pressure\_label.place(x=25,y=470, height=50,width=210)

pressure\_label1 = Label(win,text=" ",font=("Times New Roman", 17, "bold"))

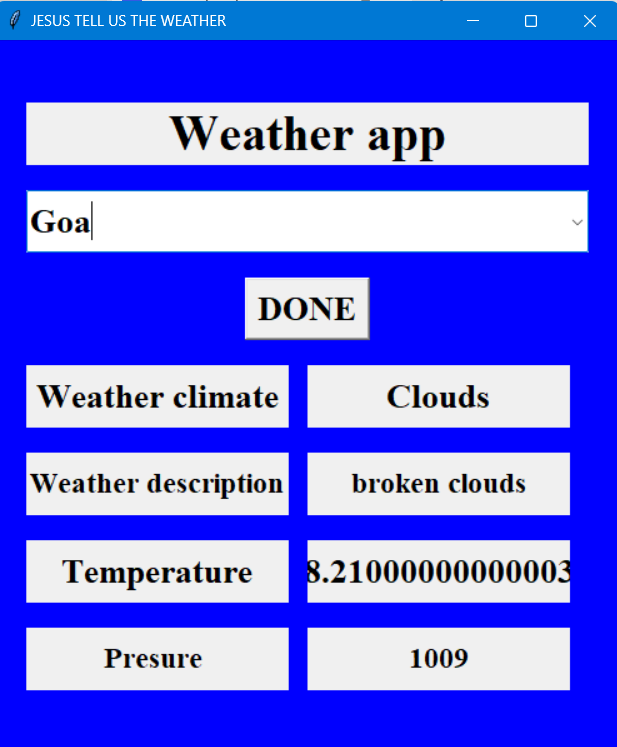
pressure\_label1.place(x=250,y=470, height=50,width=210)

done\_button = Button(win , text="DONE",font=("Times New Roman", 20, "bold"),command=data\_get)

done\_button.place(y=190, height=50,width=100,x=200 )

win.mainloop()

OUTPUT



PROJECT-3:

TITLE: QRCODE GENERATOR

CODE :

Qrcode.py

import qrcode as qr

img = qr.make("https://wcc.edu.in/")

img.save("WCC\_QR\_CODE\_GENERATOR.png")

OUTPUT:



PROJECT-4:

TITLE: Google Translator

CODE :

from tkinter import \*

from tkinter import ttk

from googletrans import Translator, LANGUAGES

def change(text="type",src="English",dest="Hindi"):

text1=text

src1=src

dest1=dest

trans = Translator()

trans1 = trans.translate(text,src=src1,dest=dest1)

return trans1.text

def data():

s = com\_sor.get()

d = com\_dest.get()

masg = sor\_txt.get(1.0,END)

textget = change(text=masg ,src=s ,dest=d)

dest\_txt.delete(1.0,END)

root = Tk()

root.title("TRANSLATOR")

root.geometry("500x600")

root.config(bg='red')

lab\_txt = Label(root, text='TRANSLATOR', font=("Times New Roman", 20, "bold"), bg="Pink")

lab\_txt.place(x=100, y=40, height=50, width=300)

frame = Frame(root).pack(side=BOTTOM)

lab\_txt1 = Label(root, text='SOURCE TEXT', font=("Times New Roman", 20, "bold"),fg="Black",bg="red")

lab\_txt1.place(x=100, y=100, height=20, width=300)

sor\_txt = Text(frame, font=("Times New Roman", 20, "bold"), wrap=WORD)

sor\_txt.place(x=10,y=130,height=150,width=480)

list\_text = list(LANGUAGES.values())

com\_sor = ttk.Combobox(frame,values=list\_text)

com\_sor.place(x=10,y=300,height=40,width=150)

com\_sor.set("english")

button\_change = Button(frame,text="TRANSLATE", relief=RAISED,command=data)

button\_change.place(x=170,y=300,height=40,width=150)

com\_dest = ttk.Combobox(frame,values=list\_text)

com\_dest.place(x=330,y=300,height=40,width=150)

com\_dest.set("english")

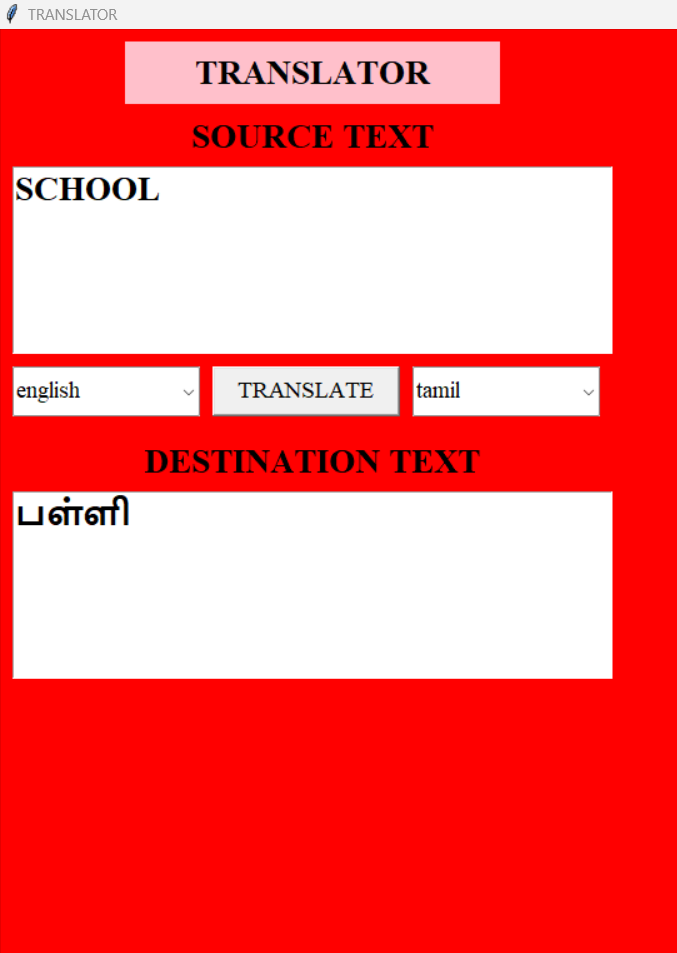
lab\_txt1 = Label(root, text='DESTINATION TEXT', font=("Times New Roman", 20, "bold"),fg="Black",bg="red")

lab\_txt1.place(x=100, y=360, height=20, width=300)

dest\_txt = Text(frame, font=("Times New Roman", 20, "bold"), wrap=WORD)

dest\_txt.place(x=10,y=400,height=150,width=480)root.mainloop()

OUTPUT



**7. CONCLUSION:**

My internship at TRIPLEM INFOTECH has been an immensely enriching experience that significantly broadened my understanding of Python programming and web development. Over the course of one month, I had the privilege to work on diverse projects, including website prediction using Flask, a weather prediction app, a QR code generator, and a Google translator. Collaborating with a talented and dynamic team, I was exposed to various aspects of software development and given substantial responsibilities that challenged me and enhanced my skills. One of the most impactful aspects of my internship was the opportunity to work alongside seasoned professionals who are experts in their fields. This hands-on involvement allowed me to gain practical experience in integrating APIs, developing user-friendly GUIs, and deploying machine learning models within web applications. The challenges I faced and the solutions I devised during these projects have deepened my understanding of both the technical and creative aspects of software development. The supportive and innovative environment at TRIPLEM INFOTECH not only honed my technical abilities but also fostered my enthusiasm for continuous learning and problem-solving. As I conclude this internship, I am empowered and inspired to leverage the knowledge and experience I have gained to contribute meaningfully to the field of software development. This transformative experience has equipped me with the skills and confidence to pursue future opportunities in technology with a passion for innovation and excellence.

**8. Web References:**

These are the web references I have used to support the information presented in this report:

* <https://www.youtube.com/watch?v=lLqEuAzh5wY>
* <https://www.youtube.com/watch?v=RsN0aXfPR1E>
* <https://www.youtube.com/watch?v=km0-GTTvt1c>
* <https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/>
* <https://www.geeksforgeeks.org/ml-linear-regression/>
* <https://www.geeksforgeeks.org/placement-prediction-app-using-flask/>
* <https://www.analyticsvidhya.com/blog/2020/09/integrating-machine-learning-into-web-applications-with-flask/>
* <https://www.geeksforgeeks.org/deploy-machine-learning-model-using-flask/>
* <https://www.geeksforgeeks.org/language-translator-using-google-api-in-python/>
* <https://pypi.org/project/googletrans/>
* <https://openweathermap.org/api>
* [https://openweathermap.org/forecast5](https://openweathermap.org/forecast5 )