**Project Report**

**On**

**FOOD COURT**

**CSC Computer Education**

**Full stack python**

**GUIDED BY: PREPARED BY:-**

**MR. Mr. R DEIVASIGAMANI, MSc., S.S.Santhiya**



**YEAR-20xx-xx**

**SUBMITTED TO**

**CSC COMPUTER EDUCATION**

**Branch : PERUNDURAI**

**64 First Floor, Amman Complex, Opp New Bus Stand,**

**Perundurai. Erode(district). 638052.**

**Email:** [**cscperundurai@gmail.com**](mailto:cscperundurai@gmail.com)**,** [**perunduraicsc@gmail.com**](mailto:perunduraicsc@gmail.com)

**Phone: 7200001338,9965553027**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Chapter** | **Title** | **Page No.** |
| **1** | **Project Overview** |  |
|  | Project Title |  |
| Project Description |
| Project Objective |
|  |
|  |
| **2** |  |  |
|  |  |  |
|  |
|  |
| **3** |  |  |
|  |  |  |
|  |
|  |
| **4** |  |  |
|  |  |  |
|  |
|  |
| **5** |  |  |
|  |  |  |
|  |
|  |
| **6** |  |  |
| **7** |  |  |
| **8** |  |  |

**1. Project Overview**

**1.1. Project Title**

**Title:Food court**

**1.2. Project Description**

**Description:**

Discover a world of delicious recipes and food inspiration.Our blog features mouthwatering dishes.Empowering your minds with healthy eating habits.Our food pamntry provides nutritious foods to families and individuals .

**Our Missions:**

* Provide high-quality,fresh and diverse food products .
* Ensure a seamless browsing experience for easy meal planning and preparation.
* Highlight sustainable and locally sourced ingredients in featured recipes.
* Foaster a community of food enthusiasts through interactive content and tips.

**1.4. Target Audience**

* **Health-consious individuals:**People who prioritize healthy eating habits and seeking organic products.
* **Busy professionals:** Searching for quick,wholesome meals and time-saving tips.
* Food enthusiasts eager to explore new cuisines and culinary techniques.

**2. Requirements**

**2.1. Functional Requirements**

* User registration and login functionality.
* High quality images.
* Social medi links.
* Contact us for suggestions & feedback.
* Responsive and mobile friendly.

**Non-Functional Requirements**

* **Performance**:The website will load within 2-3 seconds for optimal user experience.
* **Security**: User data must be protected, and password hashing should be implemented.
* **Usability**: The application should be intiative and user-friendly.
* **Reliability**: Ensure the website functions correctly accross all browsers.

**Technical Requirements**

* PROGRAMMING LANGUAGE : Python is a versatile, high-level programming language known for its simplicity and readability. Created by Guido van Rossum and first released in 1991, Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming12. Its dynamic typing and dynamic binding capabilities make it ideal for rapid application development and as a scripting language to connect existing components1. Python’s extensive standard library and the availability of numerous third-party modules make it a powerful tool for a wide range of applications, from web development and data analysis to artificial intelligence and scientific computing23. Its syntax, which emphasizes readability, allows developers to write clear and concise code, reducing the cost of program maintenance.
* FRAMEWORK : Flask is a micro web framework written in Python, designed to make web application development quick and easy. It is known for its simplicity, flexibility, and scalability, making it a popular choice among developers. Flask is based on the Werkzeug WSGI toolkit and the Jinja2 template engine, which provide the core functionalities needed for web development. Unlike larger frameworks, Flask does not include built-in database abstraction layers or form validation, allowing developers to choose their own tools and libraries as needed. This minimalistic approach makes Flask highly adaptable and suitable for both small projects and large, complex applications.
* DATABASE : A database is an organized collection of data that is stored and accessed electronically. It is designed to facilitate the efficient storage, retrieval, modification, and deletion of data. Databases are structured in a way that allows users to quickly search and retrieve specific information. This is typically achieved through a database management system (DBMS), which manages the data and provides tools for querying and reporting. Databases can be organized in various formats, such as tables with rows and columns, making it easy to process and analyze the data. They are essential for a wide range of applications, from business operations to scientific research, as they enable the systematic handling of large volumes of information.

**Frontend:**

**HTML** : HyperText Markup Language, is the standard markup language used to create and design web pages. It provides the basic structure of a webpage, which is then enhanced and modified by other technologies like CSS (Cascading Style Sheets) and JavaScript. HTML uses a series of elements, represented by tags, to enclose or wrap different parts of the content to make it appear or act in a certain way. These elements include headings, paragraphs, links, images, and more, allowing developers to build structured and visually appealing web pages. HTML is not a programming language but a markup language, meaning it is used to annotate text so that machines can understand and manipulate it.

**CSS :** CSS, or Cascading Style Sheets, is a stylesheet language used to describe the presentation of a document written in HTML or XML. It allows developers to control the layout, colors, fonts, and overall visual appearance of web pages. By separating the content from its presentation, CSS makes it easier to maintain and update web pages, as changes to the style can be made in one place and applied across multiple pages. CSS also supports responsive design, enabling web pages to adapt to different screen sizes and devices. This flexibility and efficiency make CSS an essential tool for modern web development.

**JAVASCIPT :** JavaScript is a versatile, high-level programming language primarily used to create dynamic and interactive content on web pages. It is a core technology of the World Wide Web, alongside HTML and CSS, enabling developers to implement complex features such as real-time updates, interactive maps, and animated graphics. JavaScript is an interpreted language, meaning it runs directly in the web browser without the need for prior compilation. It supports multiple programming paradigms, including object-oriented, imperative, and functional programming. This flexibility, combined with a vast ecosystem of libraries and frameworks, makes JavaScript an essential tool for both front-end and back-end web development.

**System Architecture**

**High-Level Architecture**

A diagram or description of the system architecture, including:

* **Client-side:** The user interface in the browser.
* **Server-side:** Flask application handling requests and responses.
* **Database:** Storage for user data and tasks.
* **APIs:** Any external services or APIs used.

**3.2. Components**

* **Frontend:** HTML/CSS/JavaScript templates for user interaction.
* **Backend:** Flask routes and handlers for business logic.
* **Database:** Schema and models for user data and tasks.



**4. Design**

**4.1. Database Design**

* ***Table 1:*Customer Order's Data**

Customer Name varchar(30) , Email varchar(30) ,FoodItem varchar(40) , Pincode varchar(10) , Address varchar(30).

* ***Table 2:*****Login Data**

Customer Name varchar(30) , Address varchar (50) , Email varchar(20) , Pincode varchar(10) .

* **Relationships:** How tables relate to one another (e.g., one-to-many relationship between users and tasks).
* **Schema:** Detailed schema definition for each table.

**4.2. Application Flow**

* **User Flow:** Diagram or description of how users interact with the application.
* **Request-Response Flow:** How the application processes requests and sends responses.

**5. Implementation**

**5.1. Project Setup**

* **Dependencies:** List of required libraries and tools (e.g., Flask, SQLAlchemy).
* **Installation Instructions:** Steps to set up the development environment.

**5.2. Code Structure**

* **File Organization:** Description of the file structure and organization.
* **Key Modules:** Explanation of important modules and their responsibilities.

**5.3. Example Code**

Provide snippets of critical code for key functionalities, such as user authentication or task management.

**6. Testing**

**6.1. Testing Strategy**

* **Unit Testing :** Unit testing is a software testing method where individual components or units of a program are tested to ensure they work as intended. These units, often the smallest testable parts of an application like functions or methods, are isolated from the rest of the code to verify their correctness independently. Unit tests are typically automated, allowing for quick and frequent testing during the development process1. This helps in early detection of bugs, simplifies code refactoring, and ensures that new changes do not break existing functionality. By validating each unit’s behavior, developers can maintain high code quality and reliability throughout the software development lifecycle.
* **Functional Testing :** A functional testing strategy focuses on verifying that all features work as intended. This includes testing user registration and login processes to ensure users can create accounts, log in, and recover passwords seamlessly. The search functionality must be tested for accuracy and relevance, while recipe and menu displays should be checked for completeness and correctness. The ordering process, from item selection to payment and confirmation, needs thorough testing to ensure a smooth user experience. Additionally, it’s crucial to test any integrations with third-party services, such as payment gateways and social media, to confirm they function correctly and handle errors gracefully. This strategy ensures that the website meets all functional requirements and provides a reliable user experience.
* **Integration Testing :** An integration testing strategy involves verifying that all integrated components and third-party services work together seamlessly. This includes testing the interactions between the website and external services such as payment gateways, social media platforms, and analytics tools to ensure data is correctly transmitted and received. APIs should be tested to confirm they handle requests and responses accurately and manage errors gracefully. Additionally, it’s important to check that any updates or changes in third-party services do not disrupt the website’s functionality. This strategy ensures that all integrated parts of the system work harmoniously, providing a smooth and reliable user experience.
* **End-to-End Testing :** An end-to-end testing involves validating the entire workflow from start to finish to ensure that all components and processes function correctly together. This includes testing user interactions from account creation and login to browsing recipes, placing orders, and making payments. The strategy also covers verifying the integration of third-party services like payment gateways, social media, and analytics tools.

Additionally, it involves checking the website’s performance under various conditions, ensuring data security, and confirming that all features work seamlessly across different devices and browsers. This comprehensive approach ensures that the website delivers a smooth, reliable, and secure user experience from the user’s initial visit to the final transaction.

**6.2. Tools**

* **Testing Frameworks:** pytest, unittest, etc.
* **Mocking:** Use of mock objects or services for testing.

**7. References**

List any resources, libraries, frameworks, or tools that were used or referenced during the project.

Web references

* <https://www.codeacademy.com/>
* <https://developer.mozilla.org/>
* Thompson,L.,2017.User Experience Design for websites.

Available at:

<https://www.uxdesign.com/article1234>

* Smith,J.,2010.Modern Web Development Practices.Web development journal **,**Vol**.**15(3),pp.45-67.
* Lee **,**K**.,**2015.An overview of E-commerce systems .International Journal of Bussiness **,**Vol **.**22(1)**,**pp**.**101-123**.**