

Course Name: Advanced Database Systems Design

Date of Submission: 07 May 2025

Vindhu Bhojanam - Restaurant Web Application

Team Name: Vindhu Devs

Team Members:

1. Tejaswini Bandila
2. Satya Praneeth Mallipam

Github: <https://github.com/graveinuse/Advanced-Database-Systems-Design.git>

1. Executive Summary

Vindhu Bhojanam is a full-stack web application designed to replicate a digital restaurant system, featuring dynamic menu browsing, user authentication, and a cart-based ordering experience. The main goal of the project was to build a responsive, interactive restaurant interface using Flask, SQLite, and Bootstrap with a clean UI/UX. Major deliverables include menu pages with food images and descriptions, authentication system, cart logic, checkout process, and admin dashboard.

Goals and Objectives

To develop an interactive restaurant platform where users can explore food items and add them to a personalized cart. To allow authenticated users to place orders and manage their selections using an intuitive interface. To provide an admin interface for adding, updating, and organizing menu items by category (Starters, Main Course, Biryani, Beverages, Desserts). To incorporate traditional aesthetics and Telugu content for regional identity.

Major Deliverables and Features

Fully functional user interface with login, signup, and session management. Role-based content visibility (e.g., “Add to Cart” visible only for logged-in users). A structured navigation menu and category-based food listing. Cart functionality with quantity management and total calculation. Responsive front-end using Bootstrap and custom CSS with visual flair. Integration of aesthetic elements like icons, background images, and culturally inspired visuals. Admin page for menu item entry and category classification.

Summary of Accomplishments and Challenges

The project was successfully completed with core features functioning reliably, including authentication, cart integration, dynamic routing, and database interaction. A major accomplishment was the dynamic update of the cart using session-based management, along with visual feedback (cart item count) on the navbar.

Challenges faced included:

Synchronizing cart item details across pages and ensuring consistent session data. Displaying images dynamically and ensuring image paths worked correctly in templating logic. Ensuring modularity while maintaining aesthetics and responsiveness across pages. Despite these challenges, the team collaborated effectively to meet the intended milestones and deliver a polished product ready for demonstration.

2. Project Objective

In today’s fast-paced digital world, the demand for efficient and culturally inclusive food-ordering platforms is growing rapidly. While many mainstream food delivery applications focus on a broad selection of cuisines, there remains a significant gap in platforms dedicated specifically to traditional Indian meals

and their cultural presentation. This project, **విందు భోజనం(vindhu bhojanam)**, addresses that gap by providing a seamless and personalized online ordering system designed to highlight and preserve the richness of Indian culinary traditions.

The core problem lies in the limited availability of digital platforms that offer a culturally rooted experience while also maintaining modern web functionalities such as cart management, category based browsing, and user authentication. Users often struggle to find regional food options that are well-organized, visually appealing, and easy to order online.

This project aims to serve:

- **Students** and hostel residents seek convenient and affordable home-style food.
- **Faculty** members and working professionals are interested in accessing local traditional meals without extensive effort.
- **Restaurant patrons and small business owners** aim to digitize their services in a simplified, accessible manner.

By developing a dedicated platform for ordering traditional Indian dishes—organized under categories such as Starters, Biryani, Main Course, Desserts, and Beverages, **విందు భోజనం** not only enhances customer convenience but also supports the cultural preservation of regional foods. The system simplifies the ordering experience with user-friendly features like login-based cart management, intuitive navigation, and mobile responsiveness, ensuring accessibility for all users.

Ultimately, the project bridges the gap between heritage and technology, providing a modern yet tradition-conscious platform that reflects the values of Indian hospitality in a digital format.

3. System Overview / Solution Description

The system architecture includes a front-end built with HTML, CSS, and Bootstrap, a Flask server for routing and logic, and an SQLite database for storing users, menu items, and cart sessions. Design decisions included modular HTML templates, centralized CSS styling, and minimal JS dependency.

The **Vindhu Bhojanam** system is a full-stack web application designed to provide a culturally themed online food ordering experience, primarily centered around traditional Indian dishes. It combines intuitive design with robust backend functionality to deliver a smooth user journey from browsing the menu to placing an order.

The architecture follows a **Model-View-Controller (MVC)** pattern:

Front-End (View Layer)

Built using **HTML5**, **CSS3**, and **Bootstrap 5**, the frontend is fully responsive and mobile-friendly. Each page (Home, Menu, About, Login, Sign Up, Cart, etc.) uses **modular Jinja2 templates** extending a common home.html base layout for consistency.

The UI includes interactive components such as dropdown menus, dynamic buttons (e.g., "Add to Cart"), and real-time cart item count updates based on session data.

Back-End (Controller Layer)

Flask is used as the web framework to handle routing, session management, and server-side logic. Each route (/signup, /login, /menu, /add_to_cart, /cart, /checkout, etc.) connects to appropriate Python functions and templates. Flash messaging provides real-time user feedback (e.g., login errors, successful cart addition).

Database (Model Layer)

The application uses **SQLite** for lightweight, local storage.

Four main tables are used:

1. users: Stores user credentials.
2. menu: Stores all dish data with name, category, price, description.
3. cart : Stores cart items (if persisted).
4. orders: Stores order history after successful checkout.

Design & Functional Features

User Authentication: Secure signup/login system with unique username checks.

Session-Based Cart: Items added by logged-in users are stored temporarily in the session, displayed dynamically on the cart page.

Checkout Logic: On placing the order, cart items are stored into the orders table, and the cart is cleared.

Menu Navigation: Menu items are organized into categories (Starters, Biryani, Main Course, Beverages, Desserts).

Image Handling: Each dish card includes an image fetched from the /static/images directory, passed as hidden fields during cart operations.

Development Tools Used

- Language: Python 3.x
- Web Framework: Flask
- Database: SQLite
- Frontend Libraries: Bootstrap 5, Google Fonts, Bootstrap Icons
- Editor: Visual Studio Code

- Version Control: Git (optional)

High-level System Architecture:

The application architecture is based on a client-server model. Users interact with the system through a web browser. Requests are processed by the Flask web application, which communicates with an SQLite database for data storage and retrieval.

Architecture Diagram:

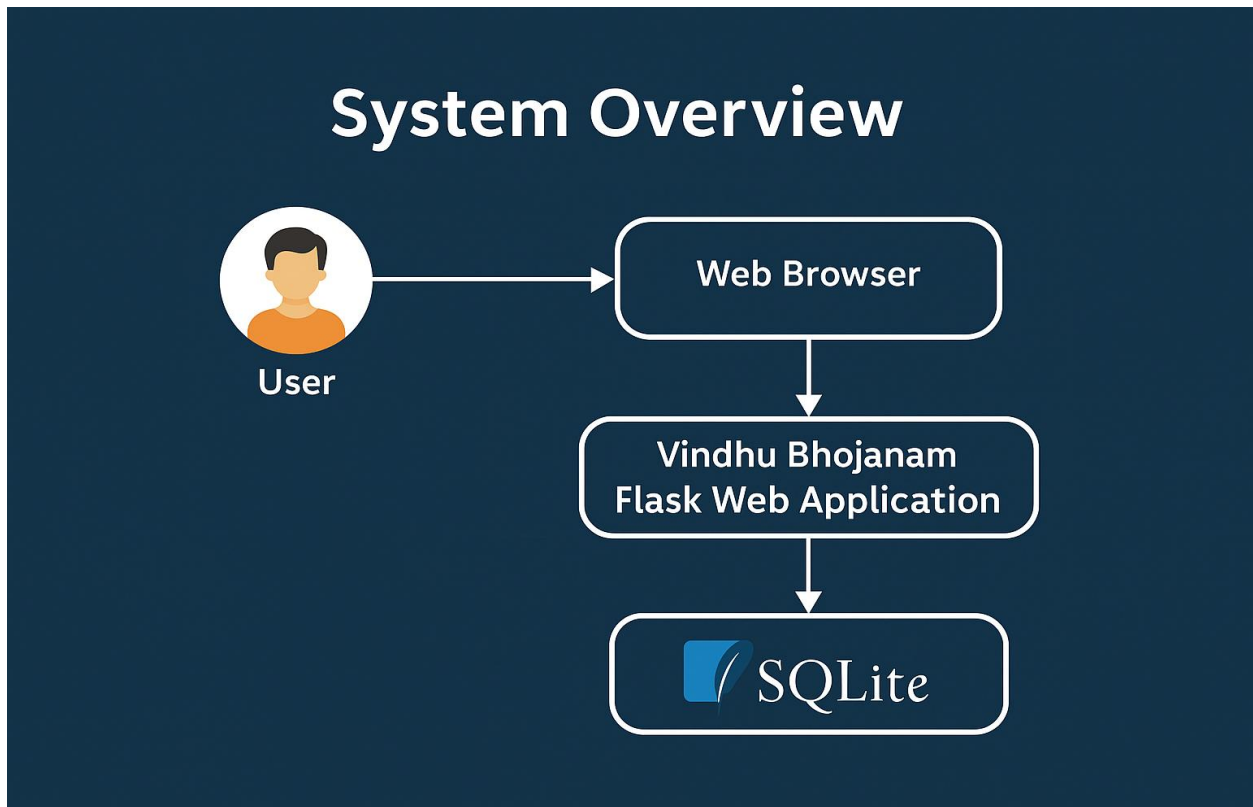


Figure 1: System Architecture Diagram, depicting interaction between User, Web Browser, Flask App, and SQLite.

Design Choices:

Flask was selected due to its simplicity and flexibility for rapid development. SQLite was chosen as a lightweight database suited for small-scale applications.

4. Development Process

The Vindhu Bhojanam project was developed using a lightweight Agile methodology, which enabled the team to make quick adjustments, take feedback into account, and build the product iteratively in stages. This approach promoted frequent cooperation, quick development cycles, and ongoing enhancement.

Methodology and Workflow : The project was broken up into several sprints, each with distinct objectives, deliverables, and due dates. We set goals like user authentication, menu rendering, and cart functionality during each four- to five-day sprint. We held a retrospective at the conclusion of each sprint in order to evaluate our progress, pinpoint any obstacles, and make plans for the future.

Every day, we had 15–20 minute in-person or virtual sync-up meetings to: Talk about the progress. Discuss any obstacles or difficulties. Assign daily tasks. Check the status of integration. The smooth integration between the frontend and backend and the avoidance of task duplication were made possible by this iterative process.

Contributions and Roles in the Team: The frontend design was **Tejaswini Bandila's** primary focus. She created responsive websites with Bootstrap, HTML, and CSS. The home, menu, about us, login, and signup pages' layouts have been implemented. Worked on the application's visual design and coherence. To capture the spirit of the culture, Telugu text elements and iconography were added. Also helped Satya praneeth to develop, manage the database and backend

Database operations and backend logic were managed by **Satya Praneeth Mallipam**. He configured the Flask web framework and took care of session management and routing. The logic for the features of user login, signup, and logout was developed. incorporated session-based cart storage, the checkout process, and the "Add to Cart" system. incorporated SQLite to store menu items and users. Update and remove functionality for cart quantities was implemented.

Tools for Collaboration:

Git was used for version control, which made it possible for us to efficiently manage branches, track changes, and combine contributions. As the primary repository, GitHub allowed both partners to push code, file issues, and settle merge disputes. Real-time conversations, feedback sharing, and document drafts were conducted using Google Docs and WhatsApp.

Significant Events among the significant turning points in development were: Initialization of the Flask application and SQLite database configuration. User verification (register, log in, and log out). Dynamic rendering and menu page design. Implementation of a cart system with quantity tracking. UI enhancements and the incorporation of responsive design, cart counters, and icons. last stage of local server deployment and testing.

5. Implementation Details

The Vindhu Bhojanam application was developed using the Flask framework as the primary backend technology, with SQLite as the database engine and HTML/CSS (enhanced with Bootstrap) for the frontend design. The project was structured with modular templates and route-driven logic to ensure a clean separation of concerns between views, server logic, and user interaction.

A critical component of the implementation was the user authentication system. Flask sessions were utilized to handle login and signup functionalities, where user credentials such as username and password were securely stored and validated using SQLite. Once authenticated, users could access restricted functionalities, including adding items to their cart and proceeding to check out.

The food menu was categorized into various sections such as Starters, Biryani, Main Course, Beverages, and Desserts. Each category had its own dynamic page template that displayed items in a card-based layout. These templates were rendered using Flask's Jinja2 templating engine, which allowed for seamless integration of data, conditional logic, and iteration directly in HTML.

Another important part of the system was the cart functionality. Once logged in, users could add menu items to their personal cart by clicking the "Add to Cart" button available on each item card. The cart data was managed within the Flask session, where each item entry included the name, price, quantity, and image path. This approach ensured persistent cart behavior throughout the session without storing cart contents in the database.

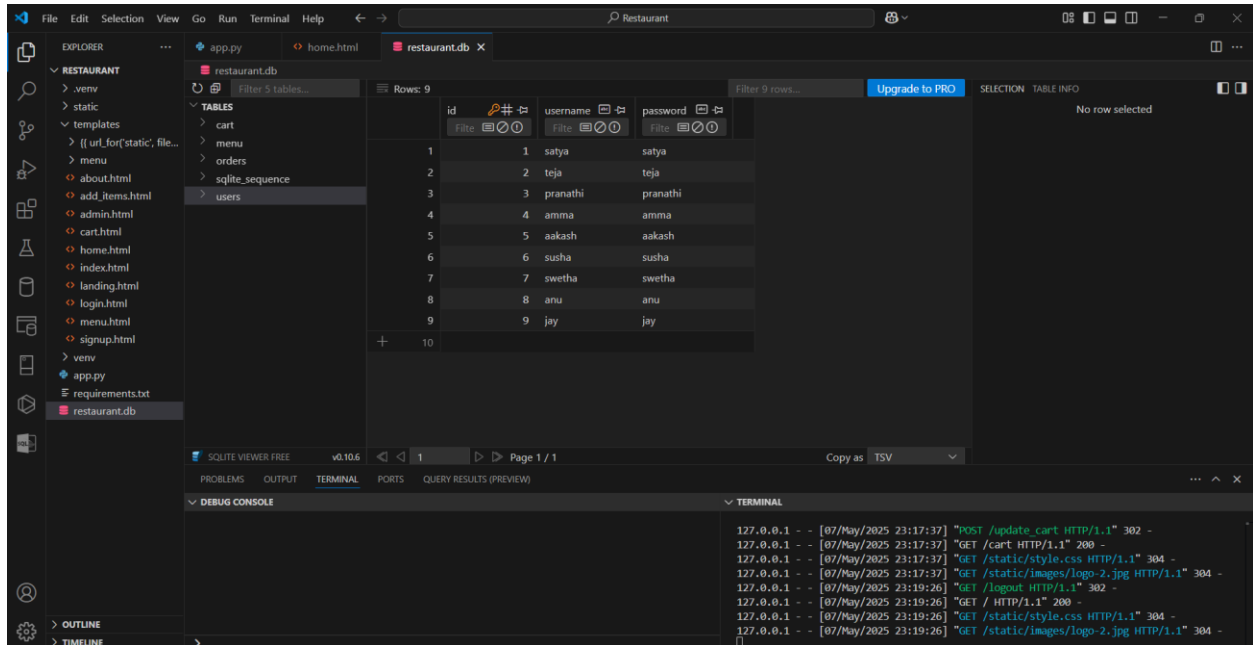
The cart page provided users with options to adjust item quantities, remove individual items, or proceed to checkout. Incrementing and decrementing quantities were handled via POST requests that updated the session data accordingly. Upon checkout, the session cart was cleared, and a confirmation message was displayed to the user.

On the backend, routes were defined in a structured manner to handle rendering, form submissions, and cart logic. Static resources such as food images and stylesheets were served from the /static directory. The database schema included two main tables: users to store authentication details, and menu to store dish-related metadata like name, category, price, and description.

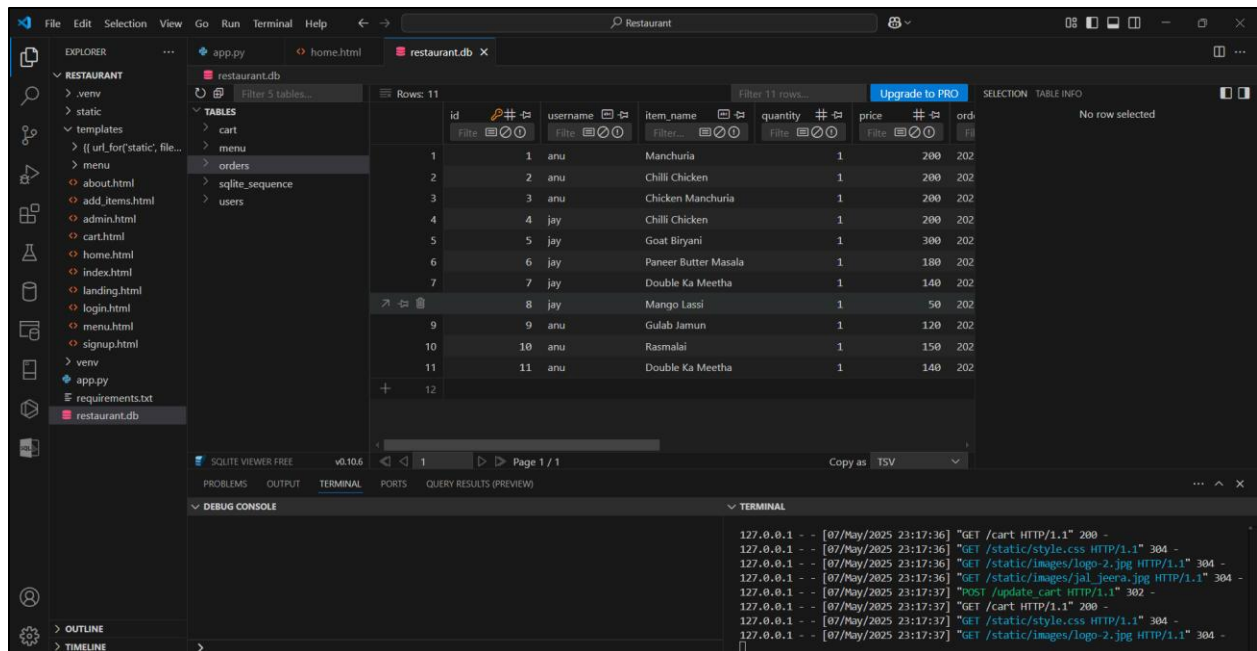
The application extensively uses CRUD operations. For instance, user sign-up and adding menu items correspond to the **Create** operation. Viewing the menu and user cart are examples of the **Read** operation. Updating item quantities in the cart is handled via the **Update** operation, and removing items corresponds to the **Delete** operation. These operations were implemented using SQLite and Flask routing mechanisms, with session data handling cart logic dynamically.

The overall codebase was kept organized with a simple directory structure that separated templates, static assets, and Python logic. This made the system scalable and maintainable as new features or categories could be easily added by extending existing routes and templates.

The below image depicts the users who have created accounts using SignUp on the home page.



- The below image shows the ordered items by the user when clicked on Checkout. The orders or the items which were in the cart during the time of checkout were present in the Database (restaurant.db)



6. Testing and Validation

Throughout the development of *Vindhu Bhojanam*, a structured manual testing approach was employed to ensure each feature worked reliably and as intended. The testing process was iterative and aligned with the development lifecycle, allowing for immediate feedback and issue resolution. Critical user interactions—such as user authentication (login and signup), cart management (adding, removing, and updating quantities of food items), and navigation between pages—were tested extensively under various conditions. These tests included verifying session persistence, checking the rendering of menu items with different user states, and validating redirection logic post login or logout actions.

To ensure cross-browser compatibility and responsive design, the application was tested on major browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge. Various screen resolutions were used to assess how the interface adapted to different devices, including laptops, tablets, and smartphones. The system’s performance remained consistent across all platforms tested.

Although automated unit testing was not implemented in this phase, thorough manual validation was conducted based on predefined functional requirements. Edge cases, such as empty cart scenarios, login failures, and missing form fields, were carefully evaluated. Stakeholder feedback from peers and instructors was also collected and used to refine user flows and interface clarity. Overall, the validation confirmed that the application met the expected functional benchmarks and user needs.

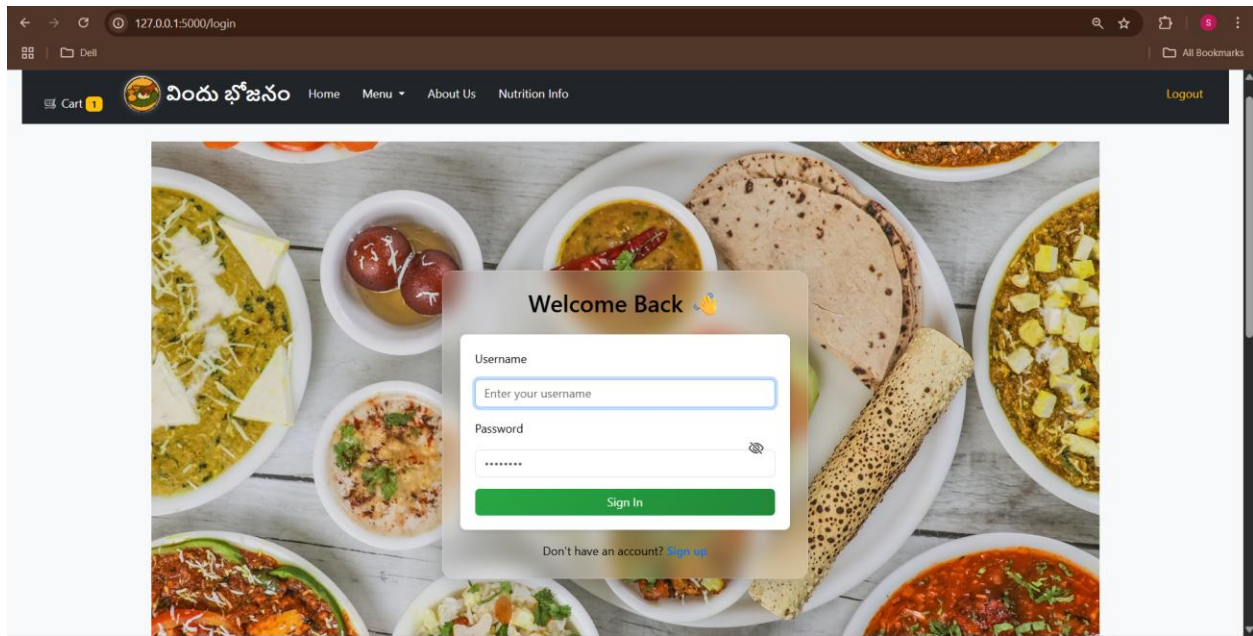
7. Results and Evaluation

The project achieved its primary objective of delivering a responsive and visually engaging web application for traditional Indian cuisine ordering. By leveraging Flask for back-end routing and session control, combined with Bootstrap for responsive design, the application demonstrated a seamless integration of design and functionality. Key modules—such as user authentication, menu categorization, cart management, and item addition—were implemented successfully and performed reliably in real-time scenarios.

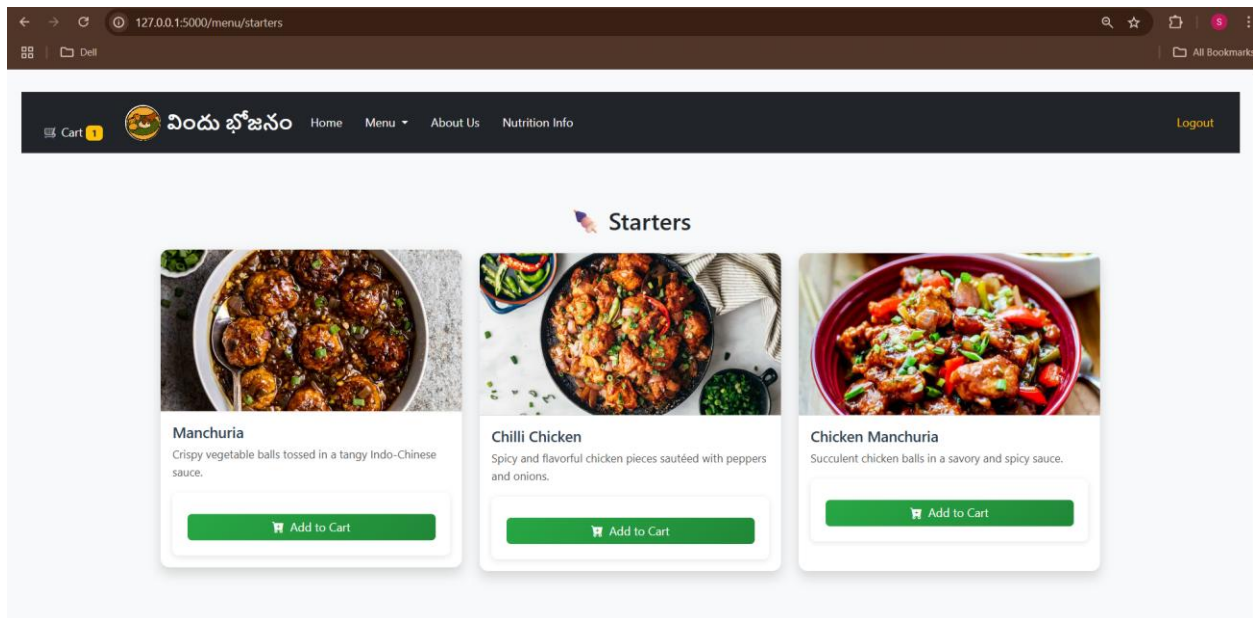
The interactive user experience was further enhanced by the inclusion of aesthetic cards, intuitive navigation, category-based filtering, and animated elements that made the web interface feel modern and immersive. User sessions were managed efficiently, enabling personalized interactions and preserving cart states throughout the browsing journey.

Feedback gathered during internal testing showed that students found the system user-friendly and appreciated its cultural context and visual presentation. The combination of structured layouts, culturally themed visuals, and dynamic logic ensured high engagement levels. Technically, the application remained stable, fast-loading, and error-free during normal operation, confirming that the system design was both effective and efficient.

HOME PAGE



MENU CARD



Biryani's



Chicken Biryani

Aromatic basmati rice layered with tender chicken and spices.

Add to Cart



Goat Biryani

Flavorful biryani made with succulent goat meat and fragrant rice.

Add to Cart



Chicken 65 Biryani

Spicy Chicken 65 pieces layered with aromatic biryani rice.

Add to Cart

Main Course



Paneer Butter Masala

Creamy tomato-based curry with soft paneer cubes, rich and flavorful.

Add to Cart



Chicken Curry

Spicy and aromatic chicken curry made with traditional Indian spices.

Add to Cart




Dal Tadka

Yellow lentils tempered with garlic, cumin and ghee — a simple classic!

Add to Cart

Your Cart






Chilli Chicken

Spicy and flavorful chicken pieces sautéed with peppers and onions.

Price: ₹200.0

Quantity: 1

Total: ₹200.0

CHECKOUT

Total: ₹200.0

 Checkout

ABOUT US



 తెలుగులో చూడండి

About Vindhu Bhojanam

"Vindhu Bhojanam" is a project based on the richness of Indian traditional cuisine. Every meal carries with it a culture and history — our mission is to bring that experience digitally.

In our web app, you can explore dishes by category, place orders the ones you love. It's a journey into the world of culinary heritage.

"Food is the form of the Supreme" 🍽️

8. Future Work and Recommendations

While the current implementation serves as a strong foundation, the application holds potential for several impactful enhancements. A critical next step is the integration of a payment gateway, such as Razorpay or Stripe, which would allow users to securely complete transactions online. This would significantly enhance the real-world applicability of the system.

Another recommended improvement is the implementation of real-time order tracking and confirmation, where users could see updates about their order status directly on their dashboard. Additionally, enabling automated email or SMS notifications for registration, successful orders, or delivery statuses would further enrich the user experience.

Security enhancements such as encrypted password storage using hashing algorithms (e.g., bcrypt) and secure session management practices are also important for scaling the platform. Features such as order history, user profile pages, and a review system for feedback could make the platform more comprehensive. Lastly, deploying the application on a public cloud platform like Heroku or AWS would ensure high availability and accessibility, enabling real-world user trials and feedback collection.

9. User Manual

To run the Vindhu Bhojanam application locally on a developer's system, users must first clone the GitHub repository containing the project source code. It is recommended to create and activate a Python virtual environment to isolate dependencies. Once the environment is activated, all required packages can be installed by executing the command `pip install -r requirements.txt` in the terminal. After setup, the application can be started by running the "app.py" script. Flask will launch the development server, and the application will be accessible via the URL `http://127.0.0.1:5000` on a browser.

The platform supports several user-facing features. After registering or logging in, users can view the categorized menu that includes Starters, Biryani, Desserts, Main Course, and Beverages. Each dish is presented in a visually appealing card layout, with options to add items to a personal cart. Users can manage the quantity of each item, remove them, or proceed to checkout.

An admin interface is also available for restaurant owners or administrators. This allows authorized users to add new dishes, categorize items, and update prices and descriptions dynamically. The entire system is designed to be intuitive for both end-users and administrators, promoting smooth interaction and efficient digital ordering workflows.

10. Appendices

Repository Link: <https://github.com/graveinuse/Advanced-Database-Systems-Design.git>

Sample Inputs: Username/Password: satya/satya

Database: SQLite

Deployment Path: `http://127.0.0.1:5000`

Function Name	Purpose
init_db()	Initializes the SQLite database and creates tables (menu, users, etc.)
signup ():	Handles user registration (Create operation)
login()	Authenticates users and starts session (Read + Session)
logout()	Ends user session
index() / landing_page()	Renders the homepage or landing page
menu() / menu_category()	Fetches and displays all menu items or by category (Read)
add_item()	Admin adds a new food item to database (Create)
add_to_cart()	Adds selected item to session-based cart (Create + Update)
cart()	Displays all items in the user's cart (Read from session)
update_cart()	Updates item quantity or removes item (Update/Delete in session)
checkout():	Stores ordered items to the database (Create in orders table)