**Project Report: Building a Basic Food Ordering System**

**Introduction:**

People are increasingly engaging with online food ordering due to its convenience and time-saving benefits, allowing them to order meals from their favorite restaurants with just a few clicks. This method offers a seamless experience by providing access to diverse menus, customer reviews, and real-time order tracking. Additionally, online food ordering platforms often feature promotions and discounts, enhancing the value and affordability of the service.

The C program me & my team build here serves as a basic food ordering system, intended to mimic the process of placing and managing food orders in a restaurant environment. This system enables users to explore a menu, submit orders, and review the total bill, showcasing essential programming principles and user interaction in C.

**System Overview:- Hery Cafe**

The food ordering system is made up of different parts that work together smoothly to give users a great experience.

**User Authentication:** Users need to log in securely or sign up for a new account to access the system's features.

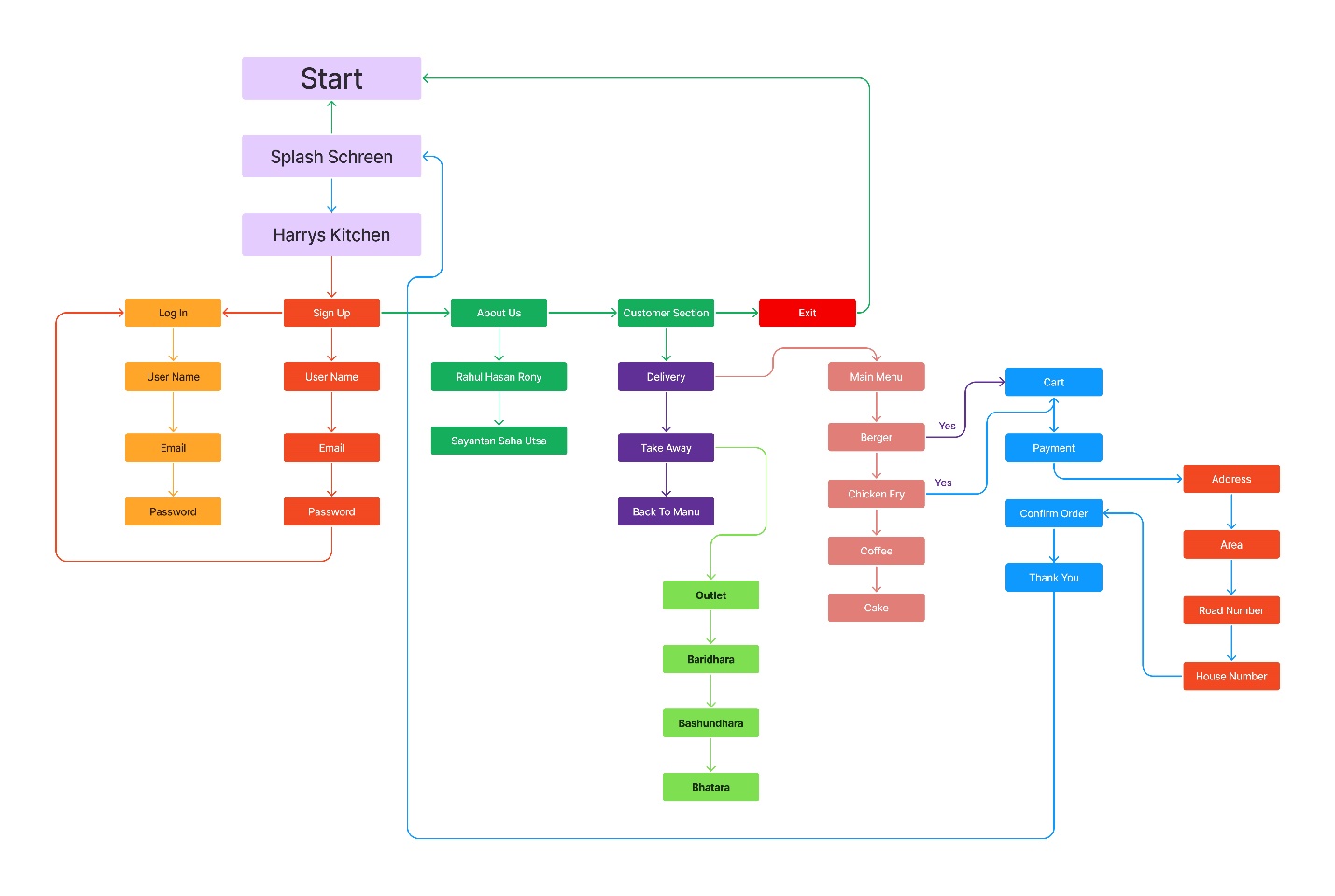
**Menu Navigation:** The system offers an organized menu, sorted into different food categories, making it simple for users to browse and choose what they want.

**Ordering Process:** Users can easily add items to their cart, customize their orders, and smoothly proceed to checkout.

**Promo Code Application:** Users can apply promo codes during checkout to get discounts on their orders, encouraging savings.

**Delivery Information:** Users can enter precise delivery addresses, ensuring their orders are promptly and accurately delivered.

**Flow Chart**



**All "Funcitons" Name Used In This Code**

1. signup: Allows users to sign up by entering their email, full name, username, and password.

It writes this information to the "info.txt" file and then calls the login function.

2. login: Allows users to log in by entering their username and password. It checks this information against what is stored in "info.txt" and calls the wrong login function if the credentials are incorrect.

3. wlogin: Similar to the login function, it handles cases where the user enters the wrong username or password.

4. mmenu: Displays the main menu options and redirects to different functions based on user input.

5. customer: Handles the customer section, allowing them to choose between delivery and takeaway.

6. oFood: Displays the menu and handles food ordering. It also manages the quantity and size of items ordered and writes this information to "counter.txt".

7. contact: Displays contact information and redirects users to login or signup based on input.

8. about: Displays information about Hery Cafe and its creators.

9. about2: Similar to the about function.

10. viewo: Displays the selected food items and allows confirmation of the order.

11. main: The main function where the program execution starts. It displays the welcome message and redirects users based on their input.

**All "Arrays" Name Used In This Code:**

1. out[70]: Stores the outlet location for takeaway orders.

2. s[100]: Array of structures storing user information (email, full name, username, password).

3. f1[100]: Array of structures storing order numbers.

4. s1[100]: Array of structures storing size and quantity of items ordered.

5. a[100]: Array of structures storing delivery address details.

6. p[50], u[50], l[50], m[50]: Arrays storing promo codes.

**All "Structures" Name Used In This Code:**

1. info: Stores user information (email, full name, username, password).

2. count: Stores order numbers.

3. quantity: Stores size and quantity of items ordered.

4. add: Stores delivery address details.

5. sum: Stores the total order amount.

**All "Files" Name Used In This Code:**

1. info.txt: Used for storing user information.

2. counter.txt: Used for counting orders.

All "Strings" Name Used In This Code: email, name, userid, pass: To store user informations .

All "Recursion" Name Used In This Code:

1. ofood - function to repeatedly handle food ordering until the user confirms the order or decides to view the order.

**"Others"**

1. int tak: Used to store the type of order (delivery or takeaway).

2. Various integer variables (i, j, k, m, n, etc.) are used for control flow and calculations.

3. Constants such as discount promo codes (u, l, m) and the delivery fee are defined.

4. Control flow statements (if, else if, else, switch) are used for decision-making throughout the code.

5. Standard input/output operations (printf, scanf) are used for user interaction and displaying information.

**Challenges Faced:**

During development, we faced various challenges:

**File Handling:** Incorporating file operations for data storage and retrieval demanded thorough error handling and data integrity checks.

**Input Validation:** Verifying user inputs to prevent errors and uphold system stability presented considerable hurdles.

**Interface Design**: Crafting a user-friendly interface that combines simplicity with functionality involved thoughtful planning of layout and interaction principles.

**Future Enhancements:**

To enhance the system further, the following improvements could be considered:

To elevate the system, consider these enhancements:

**1. Enhanced User Authentication:** Strengthen security with features like multi-factor authentication to safeguard user accounts.

**2. Graphical User Interface (GUI):** Create a visually appealing interface to improve user interaction and engagement.

**3. Database Integration:** Migrate to a database system for better scalability and performance in managing data.

**4. Order Tracking:** Enable real-time order tracking so users can monitor their orders throughout the delivery process.

**5. Payment Integration**: Integrate secure payment gateways to facilitate diverse and safe online transactions, offering users multiple payment options.

**Conclusion:**

In summary, the food ordering system offers a basic yet effective solution for users looking to order food conveniently online. While it meets its primary objectives well, there's ample room for improvement. By tackling development challenges and implementing future enhancements, the system can grow into a more robust and user-focused platform, ultimately enhancing the overall user experience.