

Software Requirements Specification (SRS)

Project: Café Fausse Website Development

1. Introduction

1.1 Purpose

This document defines the requirements for the Café Fausse Web Application.

1.2 Scope

The Café Fausse Website is a responsive, full-stack web application that provides information about the restaurant, displays the menu, allows customers to make table reservations, and showcases awards and customer testimonials. The project will be developed using React (with JSX) for the front-end, a Flask back-end, and a PostgreSQL database for persistent data management.

2. Overall Description

2.1 Product Perspective

The Café Fausse Web Application will serve as the digital front door for a fine-dining establishment. It will combine a visually appealing React-based front-end with a robust Flask back-end to handle table reservations and newsletter signup. A PostgreSQL database will ensure reliable data storage for reservations and customer information.

2.2 Product Functions

- **Web Pages:**
 - **Home Page:** Includes the restaurant's name, contact information, hours, and navigational links.
 - **Menu Page:** Displays a detailed menu segmented by categories, complete with descriptions and prices.
 - **Reservations Page:** Features a reservation form that integrates with the back-end reservation system.
 - **About Us Page:** Provides the restaurant's history, mission, and owner profiles.
 - **Gallery Page:** Showcases restaurant images, awards, and positive customer reviews.

- **Email Newsletter Signup:** A form for visitors to subscribe to a newsletter.
- **Reservation System:** A fully functional back-end that processes table bookings, checks availability, and assigns tables.

2.3 User Characteristics

- **Restaurant Customers:** Users looking to learn about the restaurant, view the menu, and book reservations.
- **Administrators/Managers:** Responsible for site content updates, menu management, and reservation oversight.
- **Development Team:** Developers building both the front-end and back-end components.

2.4 Constraints

- The front-end must be built using React and JSX.
- Styling should employ either CSS Flexbox or Grid for responsive design.
- The back-end must utilize Flask with a PostgreSQL database.
- The website must function consistently across major browsers and mobile devices.

2.5 Assumptions and Dependencies

- The site will be hosted either locally or on a publicly accessible web server.
- Users are expected to access the site using modern web browsers.
- Proper configuration of React, Flask, and PostgreSQL environments is assumed.

3. Specific Requirements

3.1 Functional Requirements

3.1.1 Home Page

- **FR-1:** Display Café Fausse's name prominently.
- **FR-2:** Show contact information and hours:
 - **Address:** 1234 Culinary Ave, Suite 100, Washington, DC 20002
 - **Phone Number:** (202) 555-4567

- **Hours:** Monday–Saturday: 5:00PM – 11:00 PM; Sunday: 5:00 PM – 9:00 PM
- **FR-3:** Include high-quality images and a consistent theme.
- **FR-4:** Provide navigation links to Menu, Reservations, About Us, and Gallery pages.

3.1.2 Menu Page

- **FR-5:** Display the menu segmented into the following categories with specific items and prices:
 - **Starters:**
 - Bruschetta – Fresh tomatoes, basil, olive oil, and toasted baguette slices: **\$8.50**
 - Caesar Salad – Crisp romaine with homemade Caesar dressing: **\$9.00**
 - **Main Courses:**
 - Grilled Salmon – Served with lemon butter sauce and seasonal vegetables: **\$22.00**
 - Ribeye Steak – 12 oz prime cut with garlic mashed potatoes: **\$28.00**
 - Vegetable Risotto – Creamy Arborio rice with wild mushrooms: **\$18.00**
 - **Desserts:**
 - Tiramisu – Classic Italian dessert with mascarpone: **\$7.50**
 - Cheesecake – Creamy cheesecake with berry compote: **\$7.00**
 - **Beverages:**
 - Red Wine (Glass) – A selection of Italian reds: **\$10.00**
 - White Wine (Glass) – Crisp and refreshing: **\$9.00**

- Craft Beer – Local artisan brews: **\$6.00**
- Espresso – Strong and aromatic: **\$3.00**

3.1.3 Reservations Page

- **FR-6:** Include a form with the following fields:
 - Time Slot (dropdown or time picker to select date and time)
 - Number of Guests
 - Customer Name
 - Email Address
 - Phone Number (optional)
- **FR-7:** Validate that the selected time slot is available and valid.
- **FR-8:** Integrate with the back-end to assign a random table (from a total of 30) when a slot is available.
- **FR-9:** Display a success message on booking or an error message if the time slot is fully booked.

3.1.4 About Us Page

- **FR-10:** Provide a detailed history of Café Fausse:
 - **About Café Fausse:** Founded in 2010 by Chef Antonio Rossi and restaurateur Maria Lopez, Café Fausse blends traditional Italian flavors with modern culinary innovation. Our mission is to provide an unforgettable dining experience that reflects both quality and creativity.
- **FR-11:** Include biographies of the founders and information on the restaurant's commitment to unforgettable dining, excellent food and locally sourced ingredients.

3.1.5 Gallery Page

- **FR-12:** Display a collection of high-resolution images that include:

- The interior ambiance of the restaurant
- Dishes from the menu
- Special events and behind-the-scenes images
- **FR-13:** Implement a lightbox feature for enlarged image viewing.
- **FR-14:** Feature awards and positive reviews:
 - **Awards:**
 - Culinary Excellence Award – 2022
 - Restaurant of the Year – 2023
 - Best Fine Dining Experience – Foodie Magazine, 2023
 - **Customer Reviews:**
 - “Exceptional ambiance and unforgettable flavors.” – Gourmet Review
 - “A must-visit restaurant for food enthusiasts.” – The Daily Bite

3.1.6 Email Newsletter Signup

- **FR-15:** Provide a form for email signup that includes input validation for proper email format.
- **FR-16:** Ensure submitted emails are stored in a backend database for future marketing purposes.

3.1.7 Reservation System (Back-end)

- **FR-17:** Develop a PostgreSQL database with the following tables:
 - **Customers Table:** Customer ID, Customer Name, Email Address, Phone Number, Newsletter Signup.
 - **Reservations Table:** Reservation ID, Customer ID, Time Slot, Table Number
- **FR-18:** Implement Flask-based logic to:

- Insert new customer records.
- Check table availability for the selected time slot.
- Assign a random table (from 30 available) if available.
- Return confirmation or error messages to the user.

3.2 Non-Functional Requirements

3.2.1 Performance Requirements

- **NFR-1:** The website should load within 3 seconds on a standard broadband connection.
- **NFR-2:** Form submissions (reservations and email sign-up) should be processed within 2 seconds.

3.2.2 Usability Requirements

- **NFR-3:** The interface must be intuitive and easy to navigate.
- **NFR-4:** The design should be consistent with the brand's identity and visually appealing.

3.2.3 Reliability Requirements

- **NFR-5:** The reservation system must maintain data integrity and prevent double or over bookings.
- **NFR-6:** The website should handle any failures in a user-friendly manner

3.2.4 Portability and Compatibility

- **NFR-7:** The application must be compatible with major browsers (Chrome, Firefox, Safari, Edge).
- **NFR-8:** The design must be responsive for desktops, tablets, and smartphones.

3.2.5 Maintainability

- **NFR-9:** The code must be modular and well-documented to facilitate future updates.

3.3 External Interface Requirements

3.3.1 User Interface

- A clean, modern, responsive interface developed with React and JSX.
- Consistent styling implemented using CSS (via Flexbox or Grid).

3.3.2 Software Interfaces

- **Front-End:** React with JSX.
- **Back-End:** Flask API endpoints for processing form submissions.
- **Database:** PostgreSQL for managing customer and reservation data.

3.3.3 Communication Interfaces

- HTTP/HTTPS protocols for client-server communications.
- RESTful API endpoints for integrating the reservation system.

4. Deployment Requirements

- The application must be deployable locally or on a web server.
- Detailed deployment instructions (including environment setup, dependency installation, and database configuration) should be provided in an accompanying README.md file.