Software Requirements Specification (SRS)

EasyPrep: Personalized Meal Planning and Grocery Delivery Web App

1. Introduction

1.1 Purpose

The purpose of **EasyPrep** is to help users plan their meals for the week and simplify grocery shopping. It allows users to create personalized meal plans and automatically generates corresponding grocery lists. This document outlines the core requirements and functionalities for developing the platform using HTML, CSS, JavaScript, and PHP.

1.2 Document Conventions

- **Abbreviations**:
    - EP: EasyPrep
    - UI: User Interface
    - DB: Database
- **Standards**:
    - This SRS follows a simplified version of IEEE SRS documentation format.

1.3 Intended Audience and Reading Suggestions

- **Developers**: Refer to Sections 3 and 4 for system features and interfaces.
- **Project Stakeholders**: Focus on Sections 1 and 2 for understanding the project scope and goals.
- **Testers**: Use functional requirements in Section 3 for test case development.

1.4 Project Scope

EasyPrep is a web-based application for creating meal plans and generating grocery lists. It allows users to:

- Choose meals for each day of the week
- View and customize auto-generated grocery lists
- Save and load their meal plans
- Optionally print or download the grocery list for shopping
  The goal is to help users eat better, save time, and stay organized.

1.5 References

- W3Schools HTML, CSS, JavaScript, PHP tutorials
- PHP PDO documentation for database interaction
- Bootstrap

2. Overall Description

2.1 Product Perspective

EasyPrep is a standalone web application with a custom backend built in PHP and a frontend using vanilla HTML, CSS, and JavaScript. It will use MySQL as the database and run on a local server (e.g., XAMPP).

2.2 Product Features

- **Home Page**: Brief about the app and how to get started
- **Meal Planner Page**: Users select meals for each day
- **Grocery List Page**: Auto-generated list based on selected meals
- **Recipe Page:** Displays a collection of recipes with details such as ingredients, cooking steps, and cuisine type.

- **Login/Sign-Up**: Simple user authentication
- **User Profile Page**: View saved plans and grocery history

2.3 User Classes and Characteristics

- **Regular Users**: Can create plans, generate lists, and manage profiles

2.4 Operating Environment

- Frontend: HTML, CSS, JavaScript
- Backend: PHP 7+
- Database: MySQL
- Server: XAMPP / Apache
- Compatible with modern web browsers on desktop and mobile

2.5 Design and Implementation Constraints

- Limited stack (no frameworks)
- Mobile responsiveness via CSS
- Lightweight performance for low-end devices

2.6 User Documentation

- On-site help section or tooltips
- Step-by-step instructions for first-time users

2.7 Assumptions and Dependencies

- Users will access the site via modern browsers
- Users will create and manage their own meals (no API integration for recipes yet)
- PHP and MySQL will be properly configured on the hosting server

3. System Features

3.1 User Authentication

- **Priority**: High
- **Description**: Users can sign up, log in, and access their saved data
- **Functional Requirements**:
    - Email and password registration/login
    - PHP session handling for login states
    - Passwords stored securely (hashed)

3.2 Meal Planner

- **Priority**: High
- **Description**: Users can assign meals to days of the week
- **Functional Requirements**:
    - Select meals for breakfast, lunch, and dinner per day
    - Meals saved to the database
    - Users can edit or delete planned meals

3.3 Grocery List Generator

- **Priority**: High
- **Description**: Auto-generates a list of ingredients from selected meals
- **Functional Requirements**:
    - PHP script to gather ingredients from selected meals
    - Avoid duplicates and group similar items
    - Option to print/download the list

3.4 Recipe Page

- **Priority:** High
- **Description:** Users can browse recipes with details such as ingredients, steps, and cuisine type
- **Functional Requirements:**
    - View list of available recipes
    - Search or filter recipes by category/cuisine
    - Display recipe details (ingredients, preparation steps)
    - Add recipe to Meal Planner or Grocery List

3.5 Save and Load Plans

- **Priority**: Medium
- **Description**: Users can save weekly plans for future use
- **Functional Requirements**:

    - Save plans to database linked to user
    - Load saved plans in the planner view

4. External Interface Requirements

4.1 User Interface

- Simple, clean interface using HTML/CSS
- Interactive meal selection using JavaScript
- Forms for login/sign-up and planning

4.2 Hardware Interfaces

- Works on laptops, desktops, tablets, and smartphones

4.3 Software Interfaces

- MySQL database for storing user data, meals, and plans
- PHP (with PDO) for server-side logic
- XAMPP or live hosting with PHP/MySQL support

4.4 Communications Interfaces

- Local server or HTTPS-enabled hosting for data protection
- Form submissions and AJAX for user actions (optional)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Pages should load within 2 seconds
- Basic caching of user plans or data (optional)

5.2 Safety Requirements

- Users' data is not shared with third parties
- Minimal personal data collection

5.3 Security Requirements

- Passwords hashed using bcrypt (or PHP password_hash)
- Secure sessions and input validation to prevent SQL injection/XSS

5.4 Software Quality Attributes

- Simple, readable codebase
- Easy to update and maintain
- Works on all major browsers

6. Other Requirements

- Optional contact form or feedback section
- Option to export meal plan as PDF

Appendices

A. Glossary

- **Meal Plan**: Weekly schedule of selected meals
- **Grocery List**: Auto-generated list of ingredients needed for a plan
- **Responsive Design**: Web design that works on all screen sizes

B. Analysis Models

- Basic flowchart of meal planning and list generation process
- Table structure of database (users, meals, plans, ingredients)

C. Issues List

- **Pending**: Add ability to reuse previous week's plan
- **Resolved**: Login and registration with PHP sessions