**Software Requirements Specification (SRS)**

**1. Introduction**

**1.1 Purpose**

The purpose of this SRS is to define the requirements for the Tax Calculation System, which will enable users to calculate different types of taxes (Income Tax, Sales Tax, and Property Tax) easily and efficiently.

**1.2 Scope**

This system will provide a web-based interface for users to enter their financial information and receive tax calculations. It will support the following features:

* Income Tax Calculation
* Sales Tax Calculation
* Property Tax Calculation
* Modern and professional user interface

**1.3 Definitions, Acronyms, and Abbreviations**

* **SRS**: Software Requirements Specification
* **UI**: User Interface
* **Tax**: A financial charge imposed by a government on individuals or entities

**1.4 References**

* N/A

**1.5 Overview**

This document will describe the overall system architecture, functionalities, and non-functional requirements of the Tax Calculation System.

**2. Overall Description**

**2.1 Product Perspective**

The Tax Calculation System will be a standalone web application accessible through any modern web browser. It will consist of a user-friendly interface that allows users to input their data and receive tax calculations in real time.

**2.2 Product Functions**

The system will provide the following functions:

1. **Income Tax Calculation**:
   * Users can input their annual income and receive the calculated income tax.
2. **Sales Tax Calculation**:
   * Users can input the purchase amount to receive the sales tax amount.
3. **Property Tax Calculation**:
   * Users can input the property value to receive the property tax amount.
4. **Display Results**:
   * After calculation, the results will be displayed in a clear and readable format.

**2.3 User Classes and Characteristics**

* **General Users**: Individuals who need to calculate their taxes. They should have basic knowledge of how to use web applications.

**2.4 Operating Environment**

* The application will run on any modern web browser (Chrome, Firefox, Safari, Edge).
* It should be compatible with both desktop and mobile devices.

**2.5 Design and Implementation Constraints**

* The application will be developed using HTML, CSS, and JavaScript.
* The system should ensure data validation and error handling for user inputs.

**2.6 User Documentation**

* User guides will be provided to help users navigate the system and utilize its features effectively.

**3. Functional Requirements**

**3.1 Income Tax Calculation**

* **Input**: User inputs their annual income.
* **Process**: Calculate the income tax based on predefined tax brackets.
* **Output**: Display the calculated income tax amount.

**3.1.1 Functional Validation**

* Ensure that the input income is a numeric value.
* The annual income must be greater than or equal to zero.

**3.2 Sales Tax Calculation**

* **Input**: User inputs the purchase amount.
* **Process**: Calculate the sales tax based on a fixed percentage (e.g., 7%).
* **Output**: Display the calculated sales tax amount.

**3.2.1 Functional Validation**

* Ensure that the purchase amount is a numeric value.
* The purchase amount must be greater than or equal to zero.

**3.3 Property Tax Calculation**

* **Input**: User inputs the property value.
* **Process**: Calculate the property tax based on a fixed rate (e.g., 1.2% of the property value).
* **Output**: Display the calculated property tax amount.

**3.3.1 Functional Validation**

* Ensure that the property value is a numeric value.
* The property value must be greater than or equal to zero.

**3.4 User Interface Requirements**

* The UI should be modern and responsive, allowing users to easily navigate through different calculators.
* Each calculator should be clearly labeled with input fields and buttons.

**3.5 Error Handling**

* If the user inputs invalid data (e.g., negative numbers, non-numeric input), the system should display an error message prompting the user to correct their input.

**4. Business Rules**

1. **Tax Bracket Rules**:
   * Income Tax is calculated based on specific brackets:
     + 10% for income up to $10,000
     + 12% for income from $10,001 to $40,000
     + 22% for income from $40,001 to $85,000
     + 24% for income from $85,001 to $160,000
     + 32% for income from $160,001 to $204,100
     + 35% for income from $204,101 to $510,300
     + 37% for income over $510,301
2. **Sales Tax Rule**:
   * Sales Tax is calculated at a fixed rate of 7% for all purchases.
3. **Property Tax Rule**:
   * Property Tax is calculated at a fixed rate of 1.2% of the property value.
4. **Validation Rule**:
   * All inputs must be validated for numeric values and must not be negative.

**5. Non-Functional Requirements**

**5.1 Performance**

* The system should provide results within 2 seconds after the user submits their data.

**5.2 Usability**

* The application should be intuitive and easy to use, allowing users to perform calculations with minimal instructions.

**5.3 Reliability**

* The system should be available 99.9% of the time and should handle errors gracefully.

**5.4 Security**

* User inputs must be validated to prevent injection attacks and ensure data integrity.

**5.5 Compatibility**

* The application must work across multiple web browsers and devices (desktop and mobile).

**6. System Requirements**

**6.1 Hardware Requirements**

* A web server to host the application.
* Client devices must have internet access.

**6.2 Software Requirements**

* Web server software (e.g., Apache, Nginx).
* Web browsers (latest versions of Chrome, Firefox, Safari, Edge).

**7. External Interfaces**

**7.1 User Interfaces**

* The application will provide a web-based UI consisting of:
  + Input forms for each type of tax calculation.
  + A results display area.
  + Navigation links to different calculators.

**7.2 Other Interfaces**

* N/A

**8. Appendices**

**8.1 Glossary**

* **Tax Bracket**: A range of incomes that are taxed at a particular rate.
* **User Interface (UI)**: The means by which the user interacts with the application.

**8.2 Issues List**

* Any potential issues identified during development should be documented and addressed promptly.