### ****Introduction****

This document defines the software requirements for the Smart Tutor Web Application. It covers functional and non-functional specifications to guide the development and implementation phases.

### ****Functional Requirements****

| **Functionality** | **Description** |
| --- | --- |
| **User Registration** | Users can register as Students or Tutors with form validation and unique email constraint. |
| **Login & Authentication** | Secure login for all user roles with SHA-256 password hashing and session management. |
| **Role Management** | Admin can assign roles and delete users. |
| **Course Management** | Tutors can post and manage their courses. Students can enroll in and view courses. |
| **Attendance Tracking** | Tutors can mark and update attendance records for their sessions. |
| **Reviews and Ratings** | Students can post reviews for tutors and rate courses. |
| **Session Control** | System maintains logged-in user data using sessions. |
| **Admin Dashboard** | Admin can view all users, courses, attendance records, and delete records as needed. |

### ****Non-Functional Requirements****

| **Requirement** | **Description** |
| --- | --- |
| **Security** | Passwords are stored securely using SHA-256 hashing. Role-based access prevents unauthorized actions. |
| **Performance** | The application should load under 3 seconds for key pages (Login, Home, Courses). |
| **Scalability** | New roles or course types can be added easily via database and role-based logic. |
| **Usability** | The user interface is responsive, form inputs are validated, and error messages guide the user. |
| **Maintainability** | Code is modular, comments are added in VB.NET backend files, and data logic is separated. |
| **Reliability** | Data operations use try-catch blocks and robust error handling to prevent crashes. |
| **Portability** | The application runs on any Windows system with .NET Framework and SQL Server. |