**Design Document**

**Assumptions**

1. **User Authentication Requirements:**
   * Users must register with an email address.
   * Passwords are hashed before storage for security.
   * A verification code is sent via email after registration.
   * Users must verify their email before logging in.
2. **Forgot Password Process:**
   * Users can reset their password via a verification code sent to their email.
   * Password reset links expire after a predefined time.
3. **Session Management:**
   * Tokens are used for authentication, stored in the user's local storage.
   * On logout, tokens and user info are cleared from the client.
4. **Admin Reports:**
   * Admin reports such as online users, registration statistics, and average login times.

**Design Decisions**

1. **Frameworks and Libraries:**
   * Backend: ASP.NET Core with MongoDB.
   * Frontend: Vue.js for user interface.
   * MongoDB Driver for database interaction.
2. **Architecture:**
   * Backend follows RESTful API design.
   * Frontend communicates with the backend via Axios HTTP requests.
3. **Data Security:**
   * Passwords are hashed using a secure hashing algorithm.
   * Tokens are signed and verified using JWT for secure session management.

**Workflow**

1. **User Registration:**
   * Users register by providing a name, surname, email, and password.
   * Backend generates a verification code and sends it via email.
2. **Email Verification:**
   * Users input the verification code to activate their account.
3. **Login:**
   * Users provide their email and password.
   * Backend validates credentials and issues a JWT token upon success.
4. **Forgot Password:**
   * Users provide their email to receive a verification code.
   * They use this code and provide a new password to reset their account.
5. **Admin** **Reports:**
   * Admins access backend APIs to retrieve reports on user activities.

**Test Scenarios**

1. **Registration Flow:**
   * Test successful registration with valid email and password.
   * Test registration with invalid email or weak password.
2. **Email Verification:**
   * Test verification with valid and invalid codes.
3. **Login Flow:**
   * Test login with correct credentials.
   * Test login with invalid email or password.
4. **Forgot Password:**
   * Test reset flow with valid and invalid verification codes.
   * Test password reset with mismatched or invalid passwords.
5. **Admin Reports:**
   * Test report generation for different date ranges and scenarios.