**General Security Features**

1. **Session-Based Access Control**: The web application uses session management to restrict user access to unauthorized pages. User sessions are started based on the login credentials (username and password).
2. **Password Security**:
   1. Passwords are securely hashed using the SHA-2 algorithm.
   2. Hashed passwords are stored in a binary32 format.
   3. Only password hashes are stored in the database, actual passwords are never retained.
3. **.htaccess Configuration**: The .htaccess file is specifically configured in the QR codes directory to prevent unauthorized access or theft of user tickets.
4. **SQL Injection Prevention**: All user input is sanitized. The application exclusively uses stored procedures for user input fields, effectively nullifying the risk of SQL injection.
5. **Restricted Registration Process**:
   1. The application does not include a self-registration feature. This is a deliberate design choice to avoid the creation of fake accounts.
   2. Registration is conducted offline at the bus ticket office, requiring verification of ID, email, etc.
6. **Ticket Validation and Security**:
   1. Tickets are uniquely identified using UUIDs and displayed as QR codes.
   2. Ticket validation is required upon bus entry and must be checked by the driver.
   3. Features like color change upon validation and a countdown timer enhance security against fraud.
   4. Ticket inspectors can check the QR code which is ticket\_id and can confirm validity.
7. **Transaction Management**:
   1. The "Purchase Ticket" button does not facilitate actual financial transactions, as this is not a live web app and integrating credit card processing poses a security risk. In a live app I would use a 3rd party payment processor such as Stripe or PayPal.
   2. Ticket purchases are recorded in a 'transactions' table, detailing user ID and ticket information (such as price), separate from the financial data.
   3. The transaction ID is linked to the ticket, indicating ownership.

**Design Features**

1. **User Interface**:
   1. The web application has a simple, minimalistic UI design for ease of use.
   2. Interactive elements like buttons are highlighted on hover or click.
   3. Accessibility is enhanced through large buttons, input fields, and text, catering to users with visual impairments or reading difficulties.
   4. Color coding and non-textual cues are employed to aid user experience and accessibility.

**Login Page**

1. **Functionality**:
   1. Features a standard login form (index.php) requiring a username and password.
   2. Includes error message handling for incorrect login credentials.

**Main Page**

1. **Ticket Management (my\_tickets.php)**:
   1. Displays all tickets owned by the user.
   2. Tickets are categorized by status (not yet valid, valid, expired), city, and type.
   3. Includes a feature for ticket validation; upon validation, the ticket's color and text change to indicate its status.
   4. A countdown timer is present to show the remaining validity period of the ticket.Tickets are sorted from valid to expired to enhance the viewing experience and user interface.
2. **Navigation**:
   1. A taskbar is included for easy navigation, featuring buttons to access 'My Tickets' and 'Buy Tickets' pages.

**Buying Tickets**

1. **Ticket Selection Process**:
   1. A dropdown menu allows users to first select the desired city.
   2. A second dropdown enables the selection of ticket type.
   3. The price of the ticket is automatically displayed once all the ticket information is selected.
2. **Ticket Purchase**:
   1. The 'Purchase Ticket' button calls procedure to create a transaction and create a ticket. Note that no actual money processing is implemented, as this is not a live project.
   2. Purchased tickets can be immediately viewed and used on the 'My Tickets' page.