

Project Report on IRCTC Ticket Reservation System

Project Title:

IRCTC Ticket Reservation System

Objective:

The aim of this project is to create a ticket reservation system for the Indian Railways (IRCTC) that allows users to manage their accounts, purchase tickets, check ticket statuses, cancel tickets, and request refunds. It integrates with a MySQL database to handle user information and ticket transactions.

Technologies Used:

- **Frontend:** Python's `tkinter` library for GUI development.
- **Backend:** MySQL database using `mysql.connector` to handle user data and ticket reservations.
- **Others:** Python libraries such as `random`, `datetime`, and `re` for generating random data, date management, and input validation.

System Features:

The system provides the following functionalities:

1. Database Connection:

- The application connects to a MySQL database for storing user information and ticket details.
- If the database does not exist, it is created along with the required tables.

2. User Account Management:

- **Account Creation:** New users can create accounts by providing personal information such as name, gender, age, date of birth, and phone number.
- **User Login:** Registered users can log in using their unique ID and password.
- **Account Settings:** After logging in, users can view their account details, delete their account, or update their personal information.

3. Ticket Management:

- **Purchase Tickets:** Logged-in users can purchase tickets by specifying the train name, date of journey, departure station, and destination station. A random PNR number is generated for each ticket.
- **Check Ticket Status:** Users can check the status of their ticket by entering the PNR number.
- **Cancel Tickets:** Users can cancel a ticket by entering the PNR number. If tickets are cancelled, refunds are initiated.

4. Main Menu:

- After login, users are presented with a main menu containing options for purchasing tickets, checking ticket status, requesting a refund, account settings, and logging out.

Functional Flow:

1. **Start Screen:** The application starts by displaying a login interface where users can either log in or create a new account.
2. **Database Connection:** Users are required to connect to the database by providing the hostname, username, and password. If the connection is successful, the system proceeds to the login screen.
3. **Account Creation:** If a user chooses to create a new account, they are assigned a random user ID and are asked to provide personal details for registration.
4. **Login:** Users enter their credentials to access their account.
5. **Main Menu:** After logging in, users can navigate to different functionalities like purchasing a ticket, checking ticket status, requesting a refund, or deleting their account.
6. **Ticket Management:** Users can purchase, check, or cancel tickets as needed.

Database Design:

The MySQL database for this system has two main tables:

1. **accounts Table:**

- Stores user account information such as ID, password, name, gender, age, date of birth, and phone number.

Schema:

```
CREATE TABLE IF NOT EXISTS accounts (  
  id INT PRIMARY KEY,  
  pass VARCHAR(16),  
  name VARCHAR(100),  
  sex CHAR(1),  
  age VARCHAR(3),  
  dob DATE,  
  ph_no CHAR(10)  
);
```

2. **tickets Table:**

- Stores information related to the tickets purchased by users, including PNR, train name, date of journey, departure and destination stations.

Schema:

```
CREATE TABLE IF NOT EXISTS tickets (  
  id INT,  
  PNR INT,  
  train VARCHAR(25),  
  doj DATE,  
  tfr VARCHAR(100),
```

```
tto VARCHAR(100)
);
```

Key Functions in the Application:

1. Database Setup:

- The application ensures that the required database and tables are created if they do not already exist.

2. Account Creation:

- Generates a random user ID for each new account.
- Validates user input and inserts the new account details into the `accounts` table.

3. User Login:

- Verifies user credentials by querying the `accounts` table and allows access to the system if authentication is successful.

4. Ticket Purchase:

- A random PNR number is generated for each ticket purchased.
- The user is prompted to enter details for the train, date of journey, departure station, and destination station.

5. Ticket Status:

- Users can enter the PNR number to check the status of their ticket. The system retrieves ticket details from the `tickets` table.

6. Ticket Cancellation:

- Users can cancel their tickets by entering the PNR number. The system deletes the ticket from the `tickets` table and processes a refund.

7. Account Deletion:

- If a user chooses to delete their account, the system first checks if they have active tickets. If there are active tickets, they can choose to cancel them. The account and associated tickets are then deleted from the database.

8. Error Handling:

- The application uses error handling techniques to manage database connection errors, invalid input, and failed transactions, providing users with appropriate error messages.

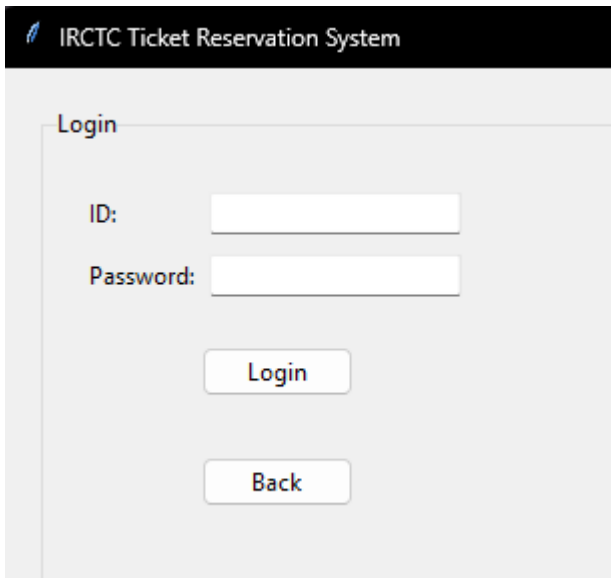
User Interface:

- The user interface is developed using the `tkinter` library. It is designed to be simple and intuitive, with each screen clearly labeled and easy to navigate.

- **Frames:** The application uses `ttk.LabelFrame` to group related widgets, such as login, account creation, and ticket management options.
- **Buttons:** Each functional button, such as "Create Account", "Login", "Purchase", and "Back", is placed within its respective frame for better usability.

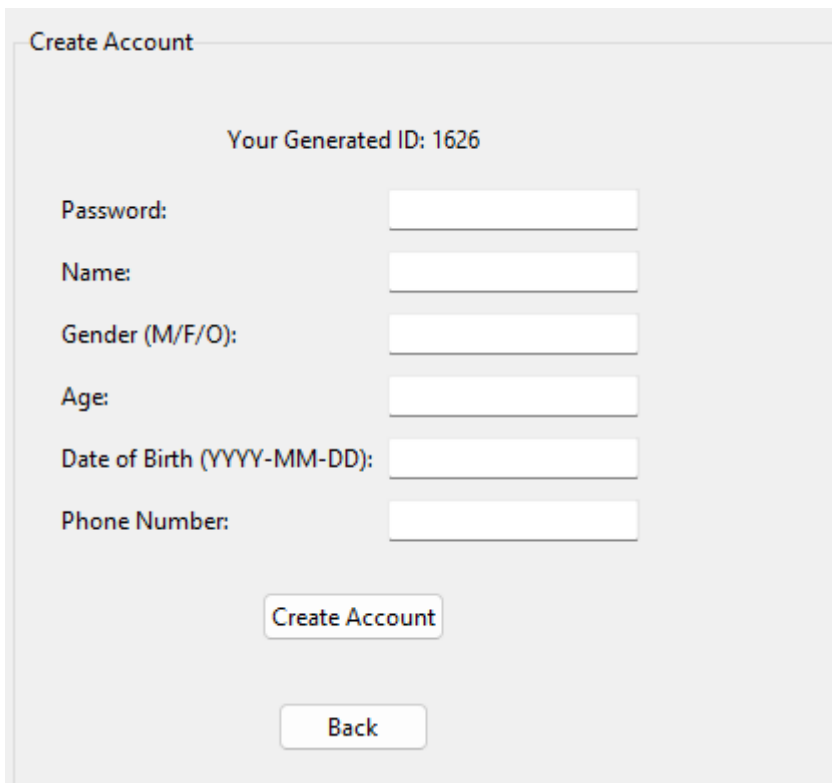
Screenshots:

- **Login Screen:** The user can input their ID and password for authentication.



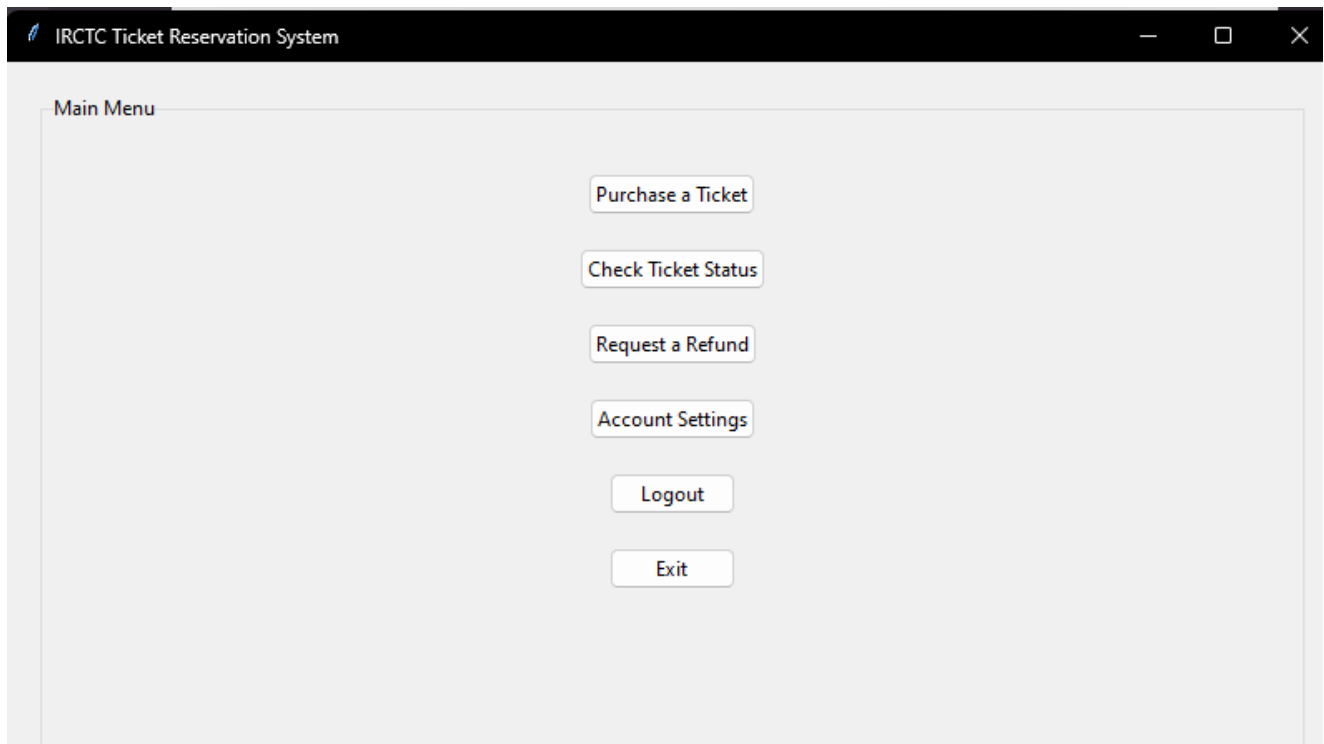
The screenshot shows the 'IRCTC Ticket Reservation System' window. Inside, there is a 'Login' section with two input fields: 'ID:' and 'Password:'. Below these fields are two buttons: 'Login' and 'Back'.

- **Account Creation Screen:** Displays fields for the user to enter their personal information, such as name, gender, age, and phone number.

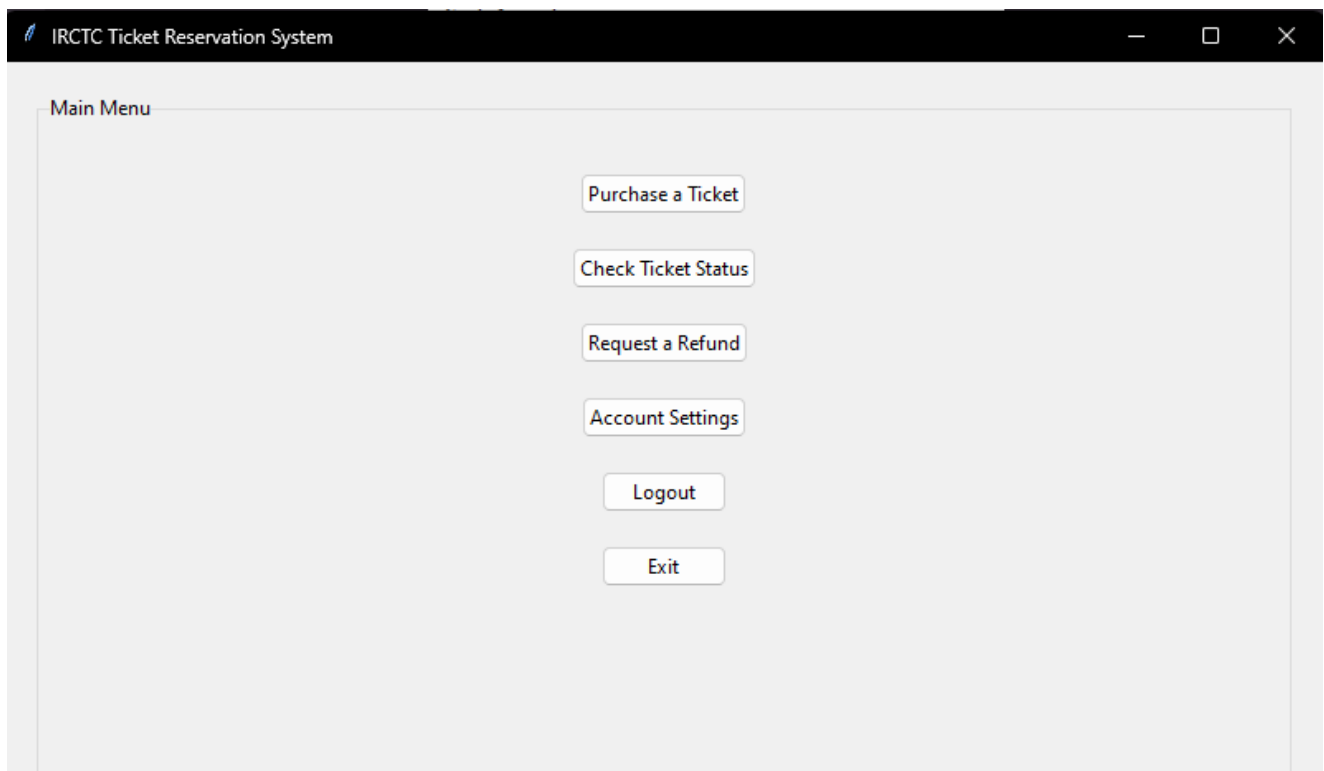


The screenshot shows the 'Create Account' window. At the top, it displays 'Your Generated ID: 1626'. Below this, there are six input fields: 'Password:', 'Name:', 'Gender (M/F/O):', 'Age:', 'Date of Birth (YYYY-MM-DD):', and 'Phone Number:'. At the bottom, there are two buttons: 'Create Account' and 'Back'.

- **Main Menu:** Provides options to purchase tickets, check ticket status, cancel tickets, or manage account settings.



- **Ticket Purchase Screen:** Allows users to input train details and generates a random PNR number.



Conclusion:

The **IRCTC Ticket Reservation System** offers a comprehensive solution for users to manage their IRCTC accounts and ticket-related activities, including purchasing tickets, checking status, and canceling tickets. The system is built with a user-friendly interface using **tkinter**, and it efficiently handles user data and ticket management using MySQL as the backend database. The project is designed to be scalable and can be extended to include more advanced features, such as train schedules, payment integration, and more robust error handling.
