David Eccles School of Business, University of Utah

IS 6495-090

Programming in Python

Final Project - Group 16

Hotel Reservation System

By,

HARSHA INJARAPU

KUSHAL TAYI

SIVA KEERTHY SURAM

Table of Contents

Introduction:

Our "Hotel Reservation System" project is a user-friendly and efficient platform designed to streamline hotel room bookings. Powered by Python, this system caters to both customers and administrators, offering a seamless reservation experience.

Customers can easily book and cancel reservations through a simple interface. They can specify their room preferences, including room type, and provide their email ID for confirmation and communication purposes. With real-time availability updates, guests can make informed decisions and secure their desired accommodations hassle-free.

The system equips administrators with a range of powerful options. They can effortlessly add, delete, and update room and reservation details, allowing for smooth management. These functionalities enable hotel staff to optimize room allocations and maintain an up-to-date inventory for maximum efficiency.

Our project incorporates an SQLite database connection to ensure efficient and secure data storage and retrieval. This database architecture allows for seamless scalability, accommodating hotels of varying sizes and accommodating future growth.

To further enhance usability, we have implemented .csv file exports, providing a convenient way to access and search reservation details. By reducing the need for repetitive physical implementation, this feature saves time and effort for both customers and administrators.

In conclusion, our "Hotel Reservation System" stands as a comprehensive solution that elevates the guest experience and streamlines hotel operations. With an intuitive user interface, powerful administrative capabilities, efficient data management, and time-saving CSV exports, our project aims to empower hotels to provide exceptional service and stay ahead in the competitive hospitality industry.

Project Approach and Strategy:

Project Overview:

A brief overview of the Hotel Reservation System project.

Clear explanation of its purpose, goals, and intended users.

Usage Guide:

Step-by-step instructions for using the system.

Clear guidelines for customers and administrators on how to perform tasks.

Database Structure:

Detailed description of the database schema and organization.

Clear explanation of tables, relationships, and data storage.

Data Dictionary:

Comprehensive dictionary of all data fields and attributes.

Clear definitions, data types, and constraints for each field.

Code Explanation:

Detailed explanation of the code's logic and functionality.

Clear insights into key algorithms or complex sections.

SQL & Database - CRUD:

The application will implement a complete CRUD (Create, Retrieve, Update, Delete) system for interacting with the database. It will involve three main tables: Hotel, Customer, and Reservation, each containing relevant information. Additionally, there will be a User Management CRUD functionality to manage user data.

1. Hotel Table:

The Hotel table will store details about various hotels, including hotel ID, name, room type, room ID, number of rooms available, and price per room.

2. Customer Table:

The Customer table will hold information about customers, such as customer ID, first name, last name, phone number, and email address.

3. Reservation Table:

The Reservation table will store data related to hotel reservations. It will include reservation ID, customer's email, hotel ID, hotel name, room type, check-in date, check-out date, and the total amount for the reservation.

4. User Management:

The User Management functionality will enable CRUD operations for managing user data. This includes creating new users with a username, encrypted password, and role (admin or regular user), retrieving user information, updating user details, and deleting user accounts.

The database will be well populated with 20 rows of data across these tables, providing ample information for testing and demonstration purposes. With this implementation, the application will efficiently manage hotel reservations, customer data, and user accounts, allowing seamless interaction with the database.

File I/O:

File I/O is utilized to handle CSV files for data population. A script is created to parse data from CSV files and store it in the database. CSV files include Customer, Hotel, and Reservation data.

Database Implementation:

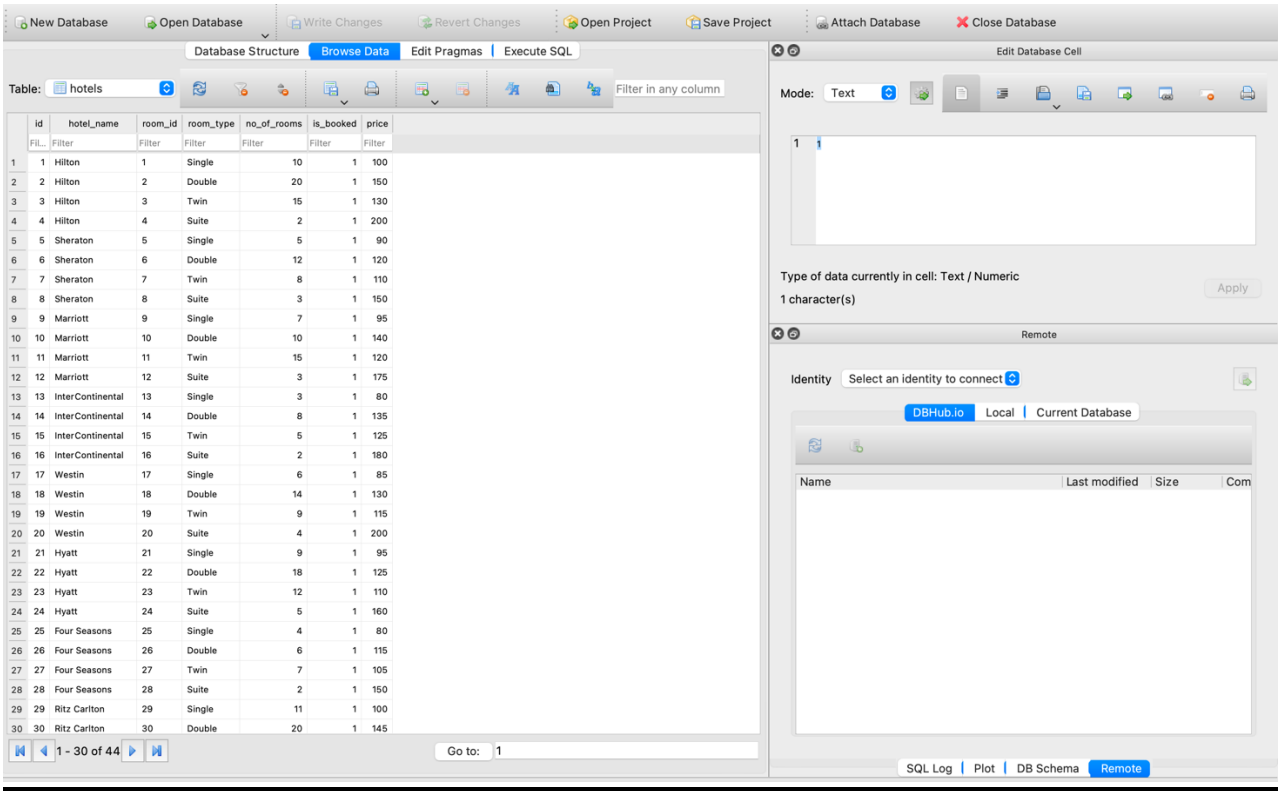
The database implementation utilizes SQLite in Python, enabling us to create, retrieve, and delete records efficiently. The foundation of our database is established through the db_base.py file, following the guidance of our professor.

Our database comprises three main tables: Hotel, Reservation, and Customer, each having its primary key (hotel_id, reservation_id, and customer_id) to support CRUD operations effectively. The admin role holds privileges to modify and customize the tables as needed.

To streamline data management, we integrated file I/O functionalities to parse CSV files containing Customer, Hotel, and Reservation data. This allows us to pre-populate the database with meaningful information. Additionally, the GUI facilitates data entry, and users can export CSV files to their local systems for reference.
The combination of SQLite as the database management system and well-structured tables with primary keys empowers us to seamlessly handle hotel reservations and customer information throughout the project.

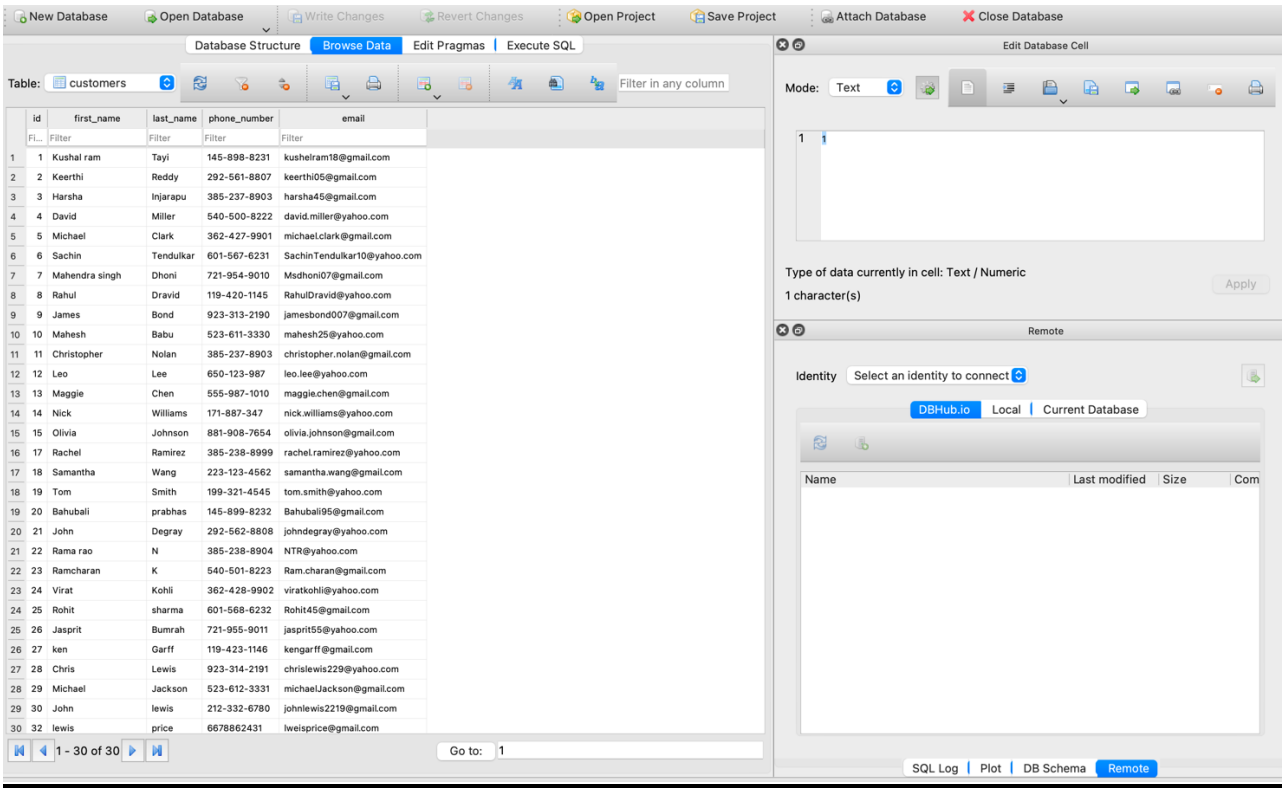
Images of Database Tables:

Hotel Table:



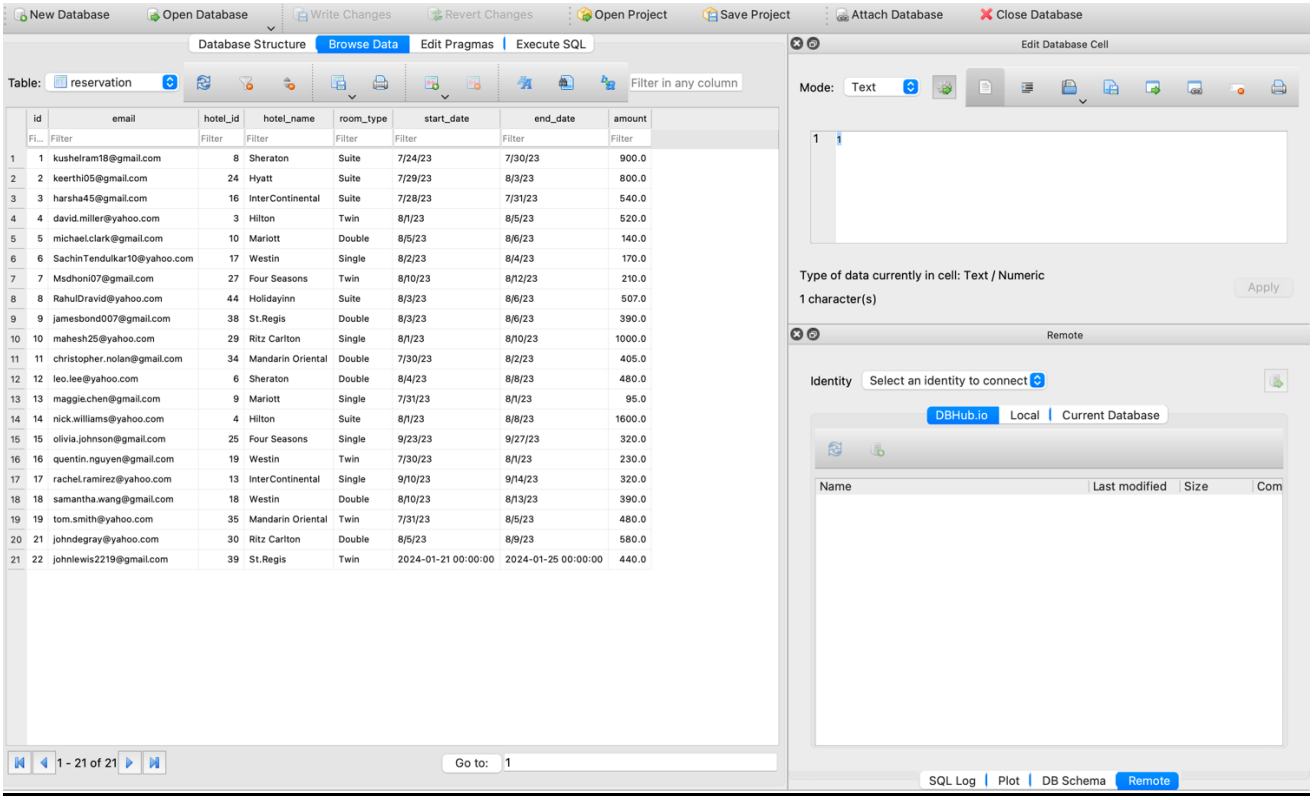
id	hotel_name	room_id	room_type	no_of_rooms	is_booked	price
1	Hilton	1	Single	10	1	100
2	Hilton	2	Double	20	1	150
3	Hilton	3	Twin	15	1	130
4	Hilton	4	Suite	2	1	200
5	Sheraton	5	Single	5	1	90
6	Sheraton	6	Double	12	1	120
7	Sheraton	7	Twin	8	1	110
8	Sheraton	8	Suite	3	1	150
9	Marriott	9	Single	7	1	95
10	Marriott	10	Double	10	1	140
11	Marriott	11	Twin	15	1	120
12	Marriott	12	Suite	3	1	175
13	InterContinental	13	Single	3	1	80
14	InterContinental	14	Double	8	1	135
15	InterContinental	15	Twin	5	1	125
16	InterContinental	16	Suite	2	1	180
17	Westin	17	Single	6	1	85
18	Westin	18	Double	14	1	130
19	Westin	19	Twin	9	1	115
20	Westin	20	Suite	4	1	200
21	Hyatt	21	Single	9	1	95
22	Hyatt	22	Double	18	1	125
23	Hyatt	23	Twin	12	1	110
24	Hyatt	24	Suite	5	1	160
25	Four Seasons	25	Single	4	1	80
26	Four Seasons	26	Double	6	1	115
27	Four Seasons	27	Twin	7	1	105
28	Four Seasons	28	Suite	2	1	150
29	Ritz Carlton	29	Single	11	1	100
30	Ritz Carlton	30	Double	20	1	145

Customer Table:



id	first_name	last_name	phone_number	email
1	Kushal ram	Tayi	145-898-8231	kushelram18@gmail.com
2	Keerthi	Reddy	292-561-8807	keerthi05@gmail.com
3	Harsha	Injarapu	385-237-8903	harsha45@gmail.com
4	David	Miller	540-500-8222	david.miller@yahoo.com
5	Michael	Clark	362-427-9901	michael.clark@gmail.com
6	Sachin	Tendulkar	601-567-6231	SachinTendulkar10@yahoo.com
7	Mahendra singh	Dhoni	721-954-9010	Msdhoni07@gmail.com
8	Rahul	Dravid	119-420-1145	RahulDravid@yahoo.com
9	James	Bond	923-313-2190	jamesbond007@gmail.com
10	Mahesh	Babu	523-611-3330	mahesh25@yahoo.com
11	Christopher	Nolan	385-237-8903	christopher.nolan@gmail.com
12	Leo	Lee	650-123-987	leo.lee@yahoo.com
13	Maggie	Chen	555-987-1010	maggie.chen@gmail.com
14	Nick	Williams	171-887-347	nick.williams@yahoo.com
15	Olivia	Johnson	881-908-7654	olivia.johnson@gmail.com
16	17 Rachel	Ramirez	385-238-8999	rachel.ramirez@yahoo.com
17	18 Samantha	Wang	223-123-4562	samantha.wang@gmail.com
18	19 Tom	Smith	199-321-4545	tom.smith@yahoo.com
19	20 Bahubali	prabhas	145-899-8232	Bahubali95@gmail.com
20	21 John	Degray	292-562-8808	johndegray@yahoo.com
21	22 Rama rao	N	385-238-8904	NTR@yahoo.com
22	23 Ramcharan	K	540-501-8223	Ram.charan@gmail.com
23	24 Virat	Kohli	362-428-9902	viratkohli@yahoo.com
24	25 Rohit	sharma	601-568-6232	Rohit45@gmail.com
25	26 Jasprit	Bumrah	721-955-9011	jasprit55@yahoo.com
26	27 ken	Garff	119-423-1146	kengarff@gmail.com
27	28 Chris	Lewis	923-314-2191	chrislewis229@yahoo.com
28	29 Michael	Jackson	523-612-3331	michaelJackson@gmail.com
29	30 John	lewis	212-332-6780	johnlewis2219@gmail.com
30	32 lewis	price	6678862431	lweisprice@gmail.com

Reservation Table:



id	email	hotel_id	hotel_name	room_type	start_date	end_date	amount
1	kushelram18@gmail.com	8	Sheraton	Suite	7/24/23	7/30/23	900.0
2	keerthi05@gmail.com	24	Hyatt	Suite	7/29/23	8/3/23	800.0
3	harsha45@gmail.com	16	InterContinental	Suite	7/28/23	7/31/23	540.0
4	david.miller@yahoo.com	3	Hilton	Twin	8/1/23	8/5/23	520.0
5	michael.clark@gmail.com	10	Mariott	Double	8/5/23	8/6/23	140.0
6	SachinTendulkar10@yahoo.com	17	Westin	Single	8/2/23	8/4/23	170.0
7	Msdhoni07@gmail.com	27	Four Seasons	Twin	8/10/23	8/12/23	210.0
8	RahulDravid@yahoo.com	44	Holidayinn	Suite	8/3/23	8/6/23	507.0
9	jamesbond007@gmail.com	38	St.Regis	Double	8/3/23	8/6/23	390.0
10	mahesh25@yahoo.com	29	Ritz Carlton	Single	8/1/23	8/10/23	1000.0
11	christopher.nolan@gmail.com	34	Mandarin Oriental	Double	7/30/23	8/2/23	405.0
12	leo.lee@yahoo.com	6	Sheraton	Double	8/4/23	8/8/23	480.0
13	maggie.chen@gmail.com	9	Mariott	Single	7/31/23	8/1/23	95.0
14	nick.williams@yahoo.com	4	Hilton	Suite	8/1/23	8/8/23	1600.0
15	olivia.johnson@gmail.com	25	Four Seasons	Single	9/23/23	9/27/23	320.0
16	quentin.nguyen@gmail.com	19	Westin	Twin	7/30/23	8/1/23	230.0
17	rachel.ramirez@yahoo.com	13	InterContinental	Single	9/10/23	9/14/23	320.0
18	samantha.wang@gmail.com	18	Westin	Double	8/10/23	8/13/23	390.0
19	tom.smith@yahoo.com	35	Mandarin Oriental	Twin	7/31/23	8/5/23	480.0
20	21	johndegray@yahoo.com	30	Ritz Carlton	Double	8/5/23	8/9/23	580.0
21	22	johnlewis2219@gmail.com	39	St.Regis	Twin	2024-01-21 00:00:00	2024-01-25 00:00:00	440.0

Object-Oriented Programming:

Object-oriented programming (OOP) is a programming paradigm that organizes code into objects, which are instances of classes. Each class defines a blueprint for creating objects with specific attributes (data members) and behaviors (methods). In the context of the hotel reservation system project, we can utilize OOP principles to create classes that represent entities such as hotels, customers, and reservations. Here's how OOP can be applied to the project:

1.Hotel Class:

Attributes: hotel_id, hotel_name, room_type, room_id, number_of_rooms, price, is_booked.

Methods: add_hotel (), update_hotel (), delete_hotel (), fetch_hotel (), fetch_hotel_by_id (), reset_database ().

2.Customer Class:

Attributes: customer_id, first_name, last_name, phone_number, email.

Methods:add_customer(),update_customer(),delete_customer(),fetch_customer(),fetch_customer_by_id(),reset_database ().

3.Reservation Class:

Attributes: reservation_id, customer_email, hotel_id, hotel_name, room_type, check_in_date, check_out_date, amount.

Methods: make_reservation (), cancel_reservation (), fetch_reservation (), reset_database ().

4. Database Class (DBbase):

Attributes: _conn (database connection), _cursor (database cursor).

Methods: __init__ (), connect (), execute_script (), get_cursor, get_connection, close_db ().

5. User Interface Class:

Attributes: None.

Methods: show menu (), handle_input (), handle_admin_input (), handle_customer_input ().

By adopting OOP, we can achieve several benefits in the project:

Modularity: Each class represents a specific entity or functionality, making the code more organized and easier to maintain.

Encapsulation: Data and methods related to each class are encapsulated within the class, promoting data privacy and abstraction.

Reusability: Once classes are defined, they can be reused in other parts of the project or in future projects with minimal modifications.

Polymorphism: Different classes can have methods with the same name, allowing for flexible and consistent usage.

For example, the Hotel class will have methods like add_hotel (), update_hotel (), delete_hotel (), etc., which will encapsulate the logic for adding, updating, and deleting hotels in the database. Similarly, the Reservation class will have methods like make_reservation () and cancel_reservation (), which will handle making new reservations and canceling existing ones.

Overall, employing OOP principles can lead to a well-structured, modular, and maintainable hotel reservation system with enhanced code readability and reusability.

User Interactive Menu:

The Hotel Reservation System provides a user-friendly interactive menu that allows both administrators and customers to navigate through the functionalities offered by the system. The menu presents different options based on the user's role and guides them through the reservation process, hotel management, and other relevant operations.

Main Menu:

The main menu serves as the entry point for users interacting with the Hotel Reservation System. It presents three options:

Admin Login: This option allows administrators to log in to the system using their credentials. After successful authentication, admins gain access to the Admin Menu, which offers various functionalities for managing hotels and reservations.

Customer Login: Customers can use this option to log in to the system using their credentials. Once logged in, they are redirected to the Customer Menu, where they can make, cancel, and view their reservations.

Admin Menu:

Upon successful admin login, the Admin Menu provides a range of options to manage hotel-related operations:

Add Hotel: Admins can add new hotels to the system by providing details such as hotel name, room type, number of rooms, and price. The system then stores this information in the Hotel table of the database.

Update Hotel: This option allows admins to modify existing hotel information, such as the hotel name, room type, number of rooms, or price.

Delete Hotel: Admins can use this option to remove hotels from the system. Deleting a hotel will also delete associated reservations, ensuring data consistency.

View All Hotels: Admins can view a list of all hotels stored in the system, displaying essential details such as hotel ID, name, room type, number of rooms, and price.

Reset Database: The reset database option allows admins to clear all data from the tables, restoring the database to its initial state.

Customer Menu:

Once customers log in successfully, they gain access to the Customer Menu, which offers functionalities related to reservations:

Make a Reservation: Customers can make a new reservation by selecting this option and providing details such as check-in date, check-out date, and hotel preferences. The system will verify the availability of rooms and calculate the total amount for the reservation.

Cancel a Reservation: Customers can cancel an existing reservation using this option. Upon cancellation, the reserved rooms become available for other customers.

Future Implementation:

In the future, we aim to enhance the Hotel Reservation System by improving the user interface to provide a more intuitive and user-friendly experience. Additionally, we plan to implement customer login functionality to personalize the booking process and enable reservation confirmation through email notifications. To facilitate seamless transactions, we will integrate secure payment gateways for secure and hassle-free payments.

Moreover, data security will be a top priority, and we will incorporate robust security measures to protect customer data from potential threats and breaches. To ensure scalability and high availability, we will deploy the application on cloud service providers like AWS, allowing us to handle increased user traffic and demand.

Furthermore, user feedback will play a crucial role in the future development of the application. We will continuously gather feedback from users and use it to make necessary improvements and enhancements to meet their evolving needs and preferences. Our goal is to create a comprehensive and efficient Hotel Reservation System that provides an exceptional experience for both customers and administrators.

Ethics and Critical Thinking:

The project will incorporate the following techniques to combine ethics and critical thinking:

Data Privacy and Security: The system will handle sensitive customer information, such as personal details and payment data. It is essential to implement robust security measures to protect this data from unauthorized access, breaches, and misuse. Adhering to data privacy laws and regulations is critical to maintain customer trust and confidence.

Fair Booking Practices: The system should prioritize fairness and transparency in booking hotel reservations. There should be no discrimination based on race, gender, nationality, or any other protected characteristics. Ensuring that all customers have equal access to booking opportunities is crucial.

Informed Consent: Customers' consent should be obtained before collecting and using their personal data for reservation purposes. Users should be provided with clear information about data usage, and they should have the option to opt-out or delete their data when needed.

Accessibility: The system should be designed to be accessible to all users, including those with disabilities. Implementing features such as screen reader compatibility and keyboard navigation will ensure that all users can use the application without any barriers.

Feedback and Continuous Improvement: Encouraging feedback from users and stakeholders will help identify ethical issues or areas of improvement. Regularly reviewing and updating the system based on feedback and emerging ethical guidelines is essential.

Conclusion:

In conclusion, the project successfully implements a hotel reservation system using Python and SQLite database. It allows customers to view available hotels, make reservations, and cancel bookings. Future enhancements may include improved UI/UX, customer login, email confirmation, payment transactions, and enhanced security. The project emphasizes privacy, fairness, and system accuracy. Constant improvements based on feedback will be made to enhance user experience and cater to industry needs. Overall, the system offers a solid foundation for further development and seamless booking experiences.