

# SMART CITY PROJECT

By Jenita Hannah P

Tagore Engineering College

Batch No: 2024-14789

Enrollment No: EBEON03251076546

# INTRODUCTION

**The Smart City Virtual Assistant** is a console-based Java application designed to serve as a digital guide for tourists and visitors in a city. Through a simple text-based interface, users can access essential services such as:

- Hotel Listings: Find the best places to stay
- Transportation Details: Information about public transport
- Air Ticket Booking: Reserve flights with ease
- Shopping: Discover the best shopping centers
- City News: Stay updated with the latest city happenings

# OBJECTIVES

- Build an interactive menu for users to select city services.
- Create modular classes to handle different services (hotels, transport, etc.).
- Simulate real-world services such as ticket booking and transport info.
- Ensure a user-friendly experience with clear input and output.
- Demonstrate core Java programming concepts like OOP and loops.

# CORE FEATURES

- Hotel Info - Provides a list of available hotels based on user input.
- Transportation - Lists transportation facilities (e.g., buses, taxis, metro) for city navigation.
- Air Ticket Booking - Allows users to book flights to the city.
- City News - Displays the latest city news for travelers to stay updated.
- Shopping - Provides details of shopping malls and markets.

# SYSTEM FLOW

- User Input: User selects an option (e.g., Hotel, Transport, etc.)
- Processing: Each service is handled by separate Java classes (e.g., HotelService, AirBookingService)
- Output: Relevant details based on user's input (e.g., booking confirmation, hotel details, transport schedule)

**User Input → [Selects service] → Processing → [Service Handler (e.g., HotelService)] → Output → [Details (e.g., booking, hotel info)]**

# TECHNOLOGIES USED

Programming Language: Java

IDE: Visual Studio Code

Core Concepts Used:

- Object-Oriented Programming (Classes, Objects, Methods)
- Scanner for Input
- Control Flow (switch, if-else)
- Execution Environment:
- Command Line Interface (CLI)

# CLASSES AND FUNCTIONS

- **Main Class (Main.java)**

Displays menu and routes to different services

- **HotelService Class**

Handles hotel-related functionality such as displaying hotels

- **AirBookingService Class**

Simulates air ticket booking process

- **ShoppingService Class**

Handles shopping-related functionality and displays major shopping malls and markets in the city

- **TransportService Class**

Provides details about available city transportation options

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Title Bar:** Main.java - SmartCityProject - Visual Studio Code
- Sidebar (Left):** EXPLORER, SMARTCITYPROJECT (containing AirBookingService.java, AirBookingService.class, Hotel.java, Hotel.class, HotelService.java, HotelService.class, Main.java, Main.class, NewsService.java, NewsService.class, ShoppingService.java, ShoppingService.class, TransportService.java, TransportService.class, TransportService.java), PROBLEMS, OUTPUT, TERMINAL, PORTS, SPELL CHECKER (7)
- Code Editor (Main Area):** The code for Main.java is displayed, which imports java.util.\* and defines a SmartCityGuide class. It includes static lists for hotels, transportation, shopping, and city news, and a static string for user name. The main method contains a while loop that calls displayMenu() and getUserChoice(). A switch statement handles user choices, with case 1 calling displayHotels() and case 2 being a break statement.
- Bottom Status Bar:** E:\SmartCityProject>[], Screen Reader Optimized, Ln 18, Col 27, Spaces: 4, UTF-8, LF, {}, Java, Go Live, Go Live, Prettier

```
import java.util.*;
class SmartCityGuide {
    static Scanner scanner = new Scanner(System.in);
    // Sample Data
    static List<String> hotels = Arrays.asList("Taj Palace", "The Oberoi", "Marriott", "Hyatt Regency", "Radisson Blu");
    static List<String> transportation = Arrays.asList("Metro", "Local Buses", "Auto Rickshaws", "Car Rentals", "Bicycles");
    static List<String> shopping = Arrays.asList("City Mall", "Street Market", "Central Bazaar", "Fashion Street");
    static List<String> cityNews = Arrays.asList("Metro Line Extension", "New Shopping Mall Opening", "Tourism Festival");
    static String userName = "Guest";
}
public static void main(String[] args) {
    while (true) {
        displayMenu();
        int choice = getUserChoice();
        switch (choice) {
            case 1:
                displayHotels();
                break;
            case 2:
        }
    }
}
```

```
E:\SmartCityProject>java Main
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -Dstderr.encoding=UTF-8

--- Welcome to Smart City Guide ---
1. Hotels
2. Transportation
3. Air Ticket Booking
4. Shopping
5. City News
6. Exit
Enter your choice: 1

--- Top Hotels in the City ---
Taj Palace - Luxury Hotel
The Oberoi - 5-Star Hotel
Marriott - International Brand Hotel
Hyatt Regency - Premium Hotel
Radisson Blu - Business Hotel

--- Welcome to Smart City Guide ---
1. Hotels
2. Transportation
3. Air Ticket Booking
4. Shopping
5. City News
6. Exit
Enter your choice: 
```

```
--- Welcome to Smart City Guide ---
1. Hotels
2. Transportation
3. Air Ticket Booking
4. Shopping
5. City News
6. Exit
Enter your choice: 3

Enter your destination: Chennai
Enter date of journey (YYYY-MM-DD): 2025-5-13
Booking ticket to Chennai on 2025-5-13...
Ticket booked successfully!

--- Welcome to Smart City Guide ---
1. Hotels
2. Transportation
3. Air Ticket Booking
4. Shopping
5. City News
6. Exit
Enter your choice: 
```

# FUTURE ENHANCEMENT

- GUI Development integrating a graphical user interface using JavaFX
- API Integration incorporating APIs to fetch real-time flight bookings, news, and hotel information
- Database Connection storing and retrieving data from a database for a dynamic, scalable experience
- Mobile Application converting the system into a mobile-friendly version for on-the-go tourists

# CONCLUSION

- The Smart City Guide project provides easy access to essential city services through the command-line interface (CLI).
- Users can interact with the console to access services such as hotel details, transport options, and shopping recommendations.
- The project has the potential to evolve, integrating advanced features and expanding to a fully functional city guide for global users, still accessible via CLI.

# THANK YOU !