BVRIT HYDERABAD
COLLEGE OF ENGINEERING FOR WOMEN

(Approved by AICTE, New Delhi and Affiliated to JNTUH,
Hyderabad) Accredited by NAAC with A Grade
Bachupally, Hyderabad-500090

Department of Computer Science & Engineering
(Artificial Intelligence & Machine Learning)

CERTIFICATE

This is to certify that the Real time research project entitled "**JAVA PROJECT(SMART C ITY)** " is a bonafide work carried out by **Ms.K.GEETHAREDDY(23WH1A6651),Ms.R.AASHRITHA(23WH1A6652), Ms. B.PRAVALIKA(23WH1A6653)**, in partial fulfillment for the award ofB. Tech degree in **Computer Science & Engineering (AI&ML), BVRIT HYDERABAD College of Engineering For Women, Bachupally, Hyderabad,**affiliated to Jawaharlal Nehru Technological University Hyderabad, under my guidance and supervision. The results embodied in the project work have not been submitted to any other.

Internal Guide Ms.
Dr.B.Lakshimi Praveena
HOD & Professor
Dept of CSE(AI&ML)

Head of the Department
Dr. B .Lakshmi
Praveena HOD &
Professor Dept of
CSE(AIML)

SMART CITY

THE

(CITY GUIDE)

AIM:

Smart City is a web-based application built using Java. It stores details of a city and displays information about the city such as hotels, shopping marts, restaurants, tourist places, transportation modes, and also some general info. This acts as a guide to the new visitors.

ABSTRACT:

Tourists and even general people travel from one place to another in order to explore or for employment purposes but before they explore, they want to get an insight data about the place. So, to help them with this, a simple city project can be the best guide for them. It is a web-based application written in Java which basically guides you about the place you’re going to visit. You can access all the details of the city. In this application, users need to sign up by entering input details and then can access all the required details of the city. It contains various modules like admin, tourism, business, and student wherein users can switch to the module as per the requirement.

DESCRIPTION:

Smart City Project - City Guide Application

* The Smart City Project is a web-based application designed to provide comprehensive information about a city, acting as a virtual guide for visitors and residents alike. Built using Java,this system aims to simplify access to city-related details, including hotels, shopping malls, restaurants, tourist attractions, transportation options, and general city information.

* This application is particularly helpful for tourists and individuals moving to a new city for work or exploration, offering them a convenient way to familiarize themselves with their surroundings. Users can register on the platform by signing up with their details, enabling them to access the city's extensive database and features.

* The project is organized into multiple modules, including Admin, Tourism, Business, and Student, allowing users to navigate and retrieve information tailored to their specific needs. Each module ensures that the platform caters to a diverse audience, providing essential and detailed insights about the city in an intuitive and user-friendly manner.

* This project serves as a one-stop solution for discovering and navigating a city efficiently, enhancing the overall travel and living experience for its users.

PROCESS/METHODOLOGY:

Process and Methodologies of the Smart City Web Application:-

The development of the Smart City web application follows a structured process and methodology to ensure that the project is efficient , scalable , and meets user requirements. The processinvolves planning, design, development, testing, and deployment , utilizing modern software development methodologies to ensure the system’s quality, reliability, and performance.

1. Requirement Gathering and Analysis:

Objective: To understand the needs of the users and stakeholders (tourists, business professionals, students, and residents).

Methodology:

Interviews/Surveys: Conducting surveys and interviews with potential users and stakeholders (city administrators, business owners, tourist agencies) to gather detailed requirements.

Use Case Analysis: Identifying different use cases (e.g., tourist etc . business professionals looking for offices) and understanding what data and functionalities need to be provided for each.

2. System Design:

Objective: To design a robust, scalable, and user-friendly architecture for the application Methodology:

Object-Oriented Design (OOD): The application is designed using object-oriented principles, which are fundamental to Java development. Key components are modeled as classes and objects to ensure modularity and reusability.

Database Design: A relational database is designed using MySQL to store user data, city information, tourism spots, business hubs, transportation modes, and other city-related information.

System Architecture: A layered architecture is typically used, separating the presentation layer, business logic layer, and data layer. This ensures a clean separation of concerns and scalability.

3. Technology Stack Selection:

Frontend:

HTML5, CSS3, JavaScript: Used for creating the user interface (UI). HTML and CSS are responsible for structuring and styling the web pages, while JavaScript adds interactivity (e.g., form validation, search filtering).

Responsive Design:

Ensuring that the application works across various screen sizes (desktops, tablets, and smartphones).

Java (J2EE):

Java is used for server-side development, ensuring platform independence and high performance. Java Servlets and JSP (Java Server Pages) are used to handle HTTP requests, business logic, and data presentation.

Database:

MYSQL:

A relational database management system used to store and manage data related to users, city information, and other modules. SQL queries are used to interact with the database and retrieve/update the data.

Frameworks/Tools:

JDBC (Java Database Connectivity): To connect the Java application to the MySQL database for fetching and storing data.

Apache Tomcat: A widely used open-source web server and servlet container for deploying Java web applications.

4. Development Process:

Agile Methodology:

The development process follows Agile principles, focusing on iterative development, flexibility, and continuous improvement. This allows for adjustments to be made based on feedback and changing requirements.

The development is broken into multiple sprints (usually 2-4 weeks each), with each sprint focusing on delivering specific features/modules (e.g., User Authentication, Admin Panel, Tourism Module).

Version Control:

Git: Used for version control to manage changes in the codebase and facilitate collaboration among team members. GitHub or GitLab repositories are commonly used for code storage and management.

Module Development:

User Authentication:

A secure system is developed for user registration, login, and profile management.

Admin Module: The admin panel is designed to allow the city administrators to manage and update data related to hotels, restaurants, and tourist attractions.

5. Testing:

Unit tests are created for individual components or modules of the application to ensure that each part functions correctly. Java’s JUnit framework can be used for this purpose.

Ensures that all components work together as expected, such as verifying that user authentication works properly with the database, or that the search and filter functionalities fetch the correct results.

Tests the entire system to ensure that all the modules (tourism, business, student, admin) function as intended when integrated together.

Real-world users are involved in testing the application. Feedback is gathered to identify any issues or improvements needed to enhance the user experience.

Load testing is done to ensure that the application can handle multiple simultaneous users without crashing, especially in a live production environment.

6. Deployment:

The application is deployed to a web server such as Apache Tomcat or Jetty, which handles incoming requests and serves the content to
users.MySQL database is set up on the server, and data migration is done to populate the database with city-related information. If scaling is required, the application can be deployed on cloud platforms like AWS or Google Cloud for easy scalability and high availability.

7. Maintenance and Updates:

Bug Fixes and Patches: Regular updates are done to fix bugs, patch security vulnerabilities, and ensure smooth operation of the application.

User Feedback Integration: Based on user feedback and analytics, new features or modifications are implemented to enhance the application further.

Content Updates: City information, events, and other dynamic data are regularly updated to keep the content fresh and relevant.

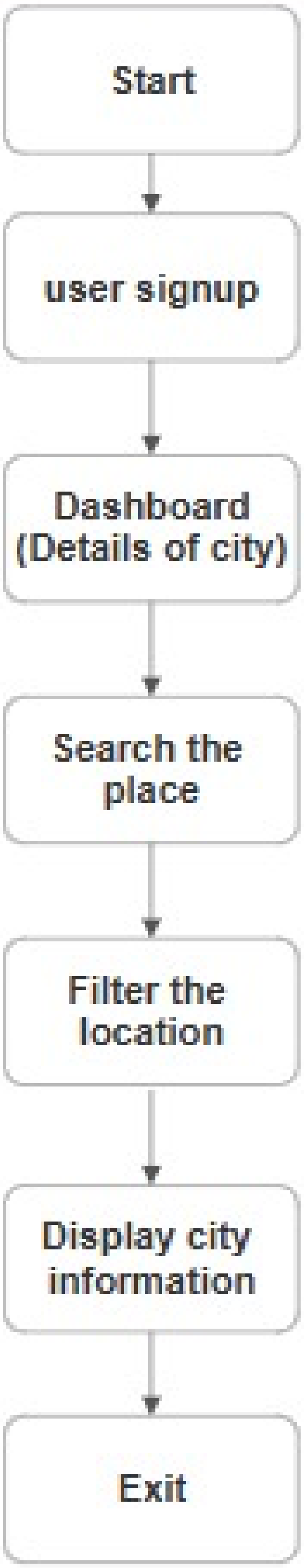
Agile Development: The iterative and flexible approach of Agile allows teams to respond to changes quickly, making it ideal for projects where requirements may evolve over time. Development cycles are divided into short sprints, with regular reviews and updates.

Object-Oriented Development: Java's object-oriented principles (encapsulation, inheritance, polymorphism) are used to design the application in a modular and reusable way, making it easier to maintain and extend.

Waterfall Model: While Agile is the primary methodology, for certain parts like system design or testing, a Waterfall approach may be used to ensure that clear documentation and structured planning are in place before moving to the next phase.

Iterative Development: Each module of the application is developed iteratively, with user feedback incorporated at each stage to ensure that the final product meets user needs.

FLOW CHART:


Start
user signup
Dashboard (Details of city)
Search the place
Filter the location
Display city information
Exit

PROGRAM CODE:

```
import java.awt.*;
import javax.swing.*;

public class SmartCityApp {
public static void main(String[] args) {
new SignUpPage();
}
}


class SignUpPage extends JFrame {
private JTextField nameField, emailField, cityField;
private JButton signUpButton;

public SignUpPage() {

setTitle("Sign Up");
setSize(240, 150); // Reduced size
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new GridLayout(4, 2, 3, 3)); // Compact spacing

JLabel nameLabel = new JLabel("Name:");
```

```
nameLabel.setHorizontalAlignment(SwingConstants.CENTER);
add(nameLabel);

nameField = new JTextField();
nameField.setFont(new Font("Arial", Font.PLAIN, 10));
add(nameField);

JLabel emailLabel = new JLabel("Email:");
emailLabel.setHorizontalAlignment(SwingConstants.CENTER);
add(emailLabel);

emailField = new JTextField();
emailField.setFont(new Font("Arial", Font.PLAIN, 10));
add(emailField);

JLabel cityLabel = new JLabel("City:");
cityLabel.setHorizontalAlignment(SwingConstants.CENTER);
add(cityLabel);

cityField = new JTextField();
cityField.setFont(new Font("Arial", Font.PLAIN, 10));
add(cityField);

signUpButton = new JButton("Sign Up");
signUpButton.setFont(new Font("Arial", Font.BOLD, 10));
add(new JLabel()); // Spacer
add(signUpButton);
```

```
signUpButton.addActionListener(e -> {
String name = nameField.getText();
String city = cityField.getText();

if (name.isEmpty() || city.isEmpty()) {
JOptionPane.showMessageDialog(this, "Please fill all fields",
"Error", JOptionPane.ERROR_MESSAGE);
} else {

new Dashboard(name, city);
dispose();
}
});

setVisible(true);
}
}


class Dashboard extends JFrame
{
public Dashboard(String userName, String city)
{

setTitle("Dashboard");
setSize(320, 200);

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new BorderLayout(5, 5));
```

```
JLabel welcomeLabel = new JLabel("Welcome, " + userName + "!
Current City: " + city, JLabel.CENTER);

welcomeLabel.setFont(new Font("Arial", Font.BOLD, 12));
add(welcomeLabel, BorderLayout.NORTH);



JPanel buttonPanel = new JPanel(new GridLayout(1, 3, 3, 3));



JButton businessButton = new JButton("Business");
JButton studentButton = new JButton("Students");
JButton touristButton = new JButton("Tourists");

businessButton.setFont(new Font("Arial", Font.PLAIN, 10));
studentButton.setFont(new Font("Arial", Font.PLAIN, 10));
touristButton.setFont(new Font("Arial", Font.PLAIN, 10));

buttonPanel.add(businessButton);
buttonPanel.add(studentButton);
buttonPanel.add(touristButton);
add(buttonPanel, BorderLayout.CENTER);
```

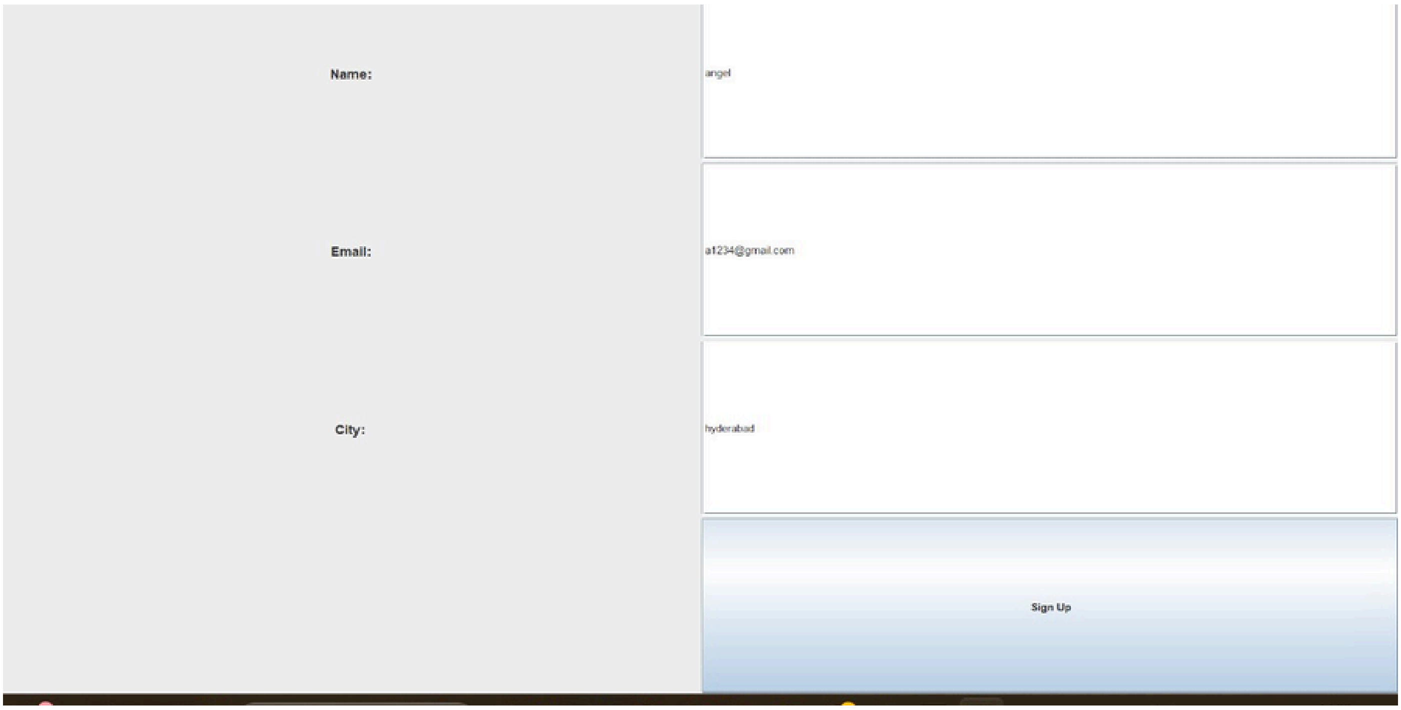
```
businessButton.addActionListener(e -> showPlaces("Business
People", new String[]{"Conference Centers", "Co-working Spaces",
"Hotels"}));
studentButton.addActionListener(e -> showPlaces("Students", new
String[]{"Libraries", "Universities", "Cafes"}));
touristButton.addActionListener(e -> showPlaces("Tourists", new
String[]{"Museums", "Parks", "Historical Monuments"}));

JButton exitButton = new JButton("Exit");
exitButton.setFont(new Font("Arial", Font.BOLD, 10));
exitButton.addActionListener(e -> System.exit(0)); // Exit the program
add(exitButton, BorderLayout.SOUTH);

setVisible(true);
}


private void showPlaces(String category, String[] places)
{
StringBuilder message = new StringBuilder("Places for " + category +
":\n");
for (String place : places) {
message.append("- ").append(place).append("\n");
}
JOptionPane.showMessageDialog(this, message.toString(), category
+ " Places", JOptionPane.INFORMATION_MESSAGE);
}
}
```

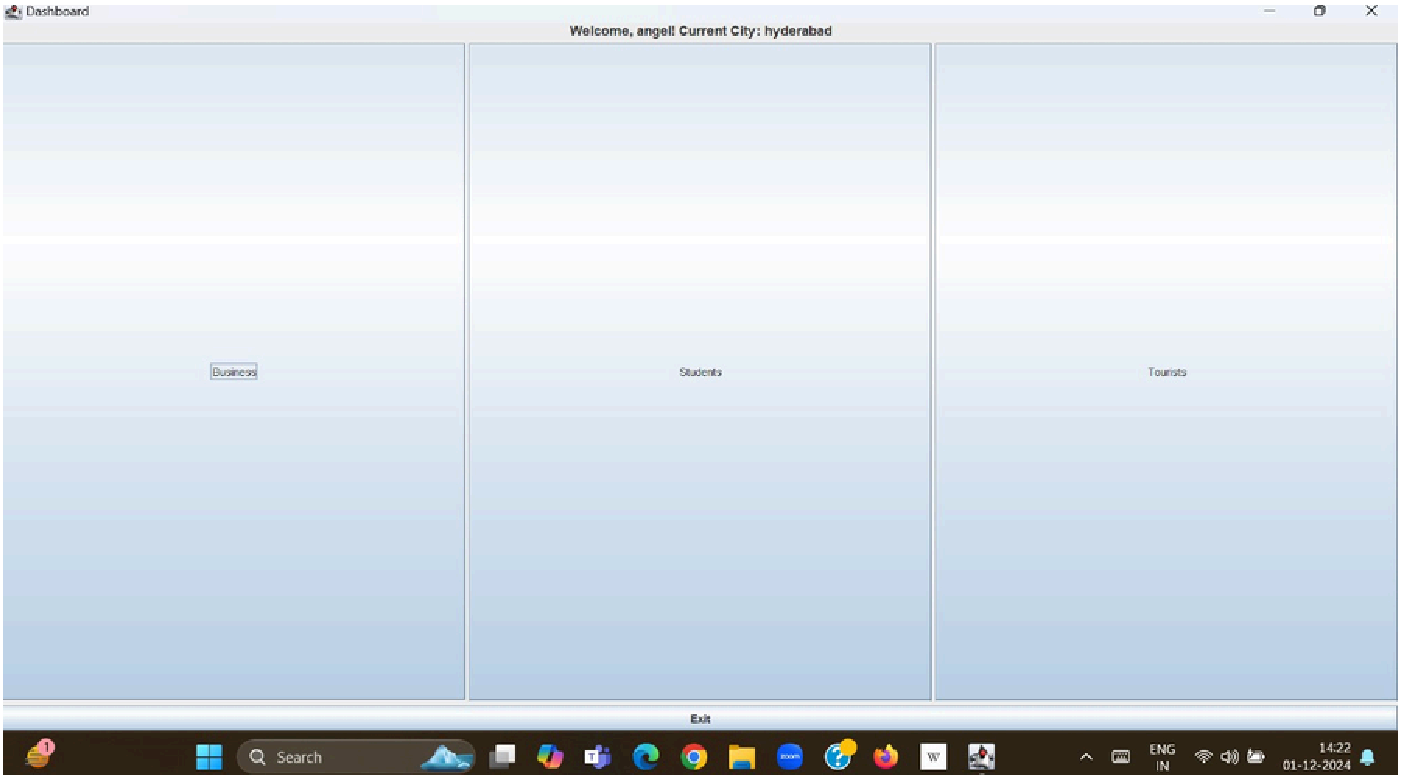
OUTPUT SCREENS:



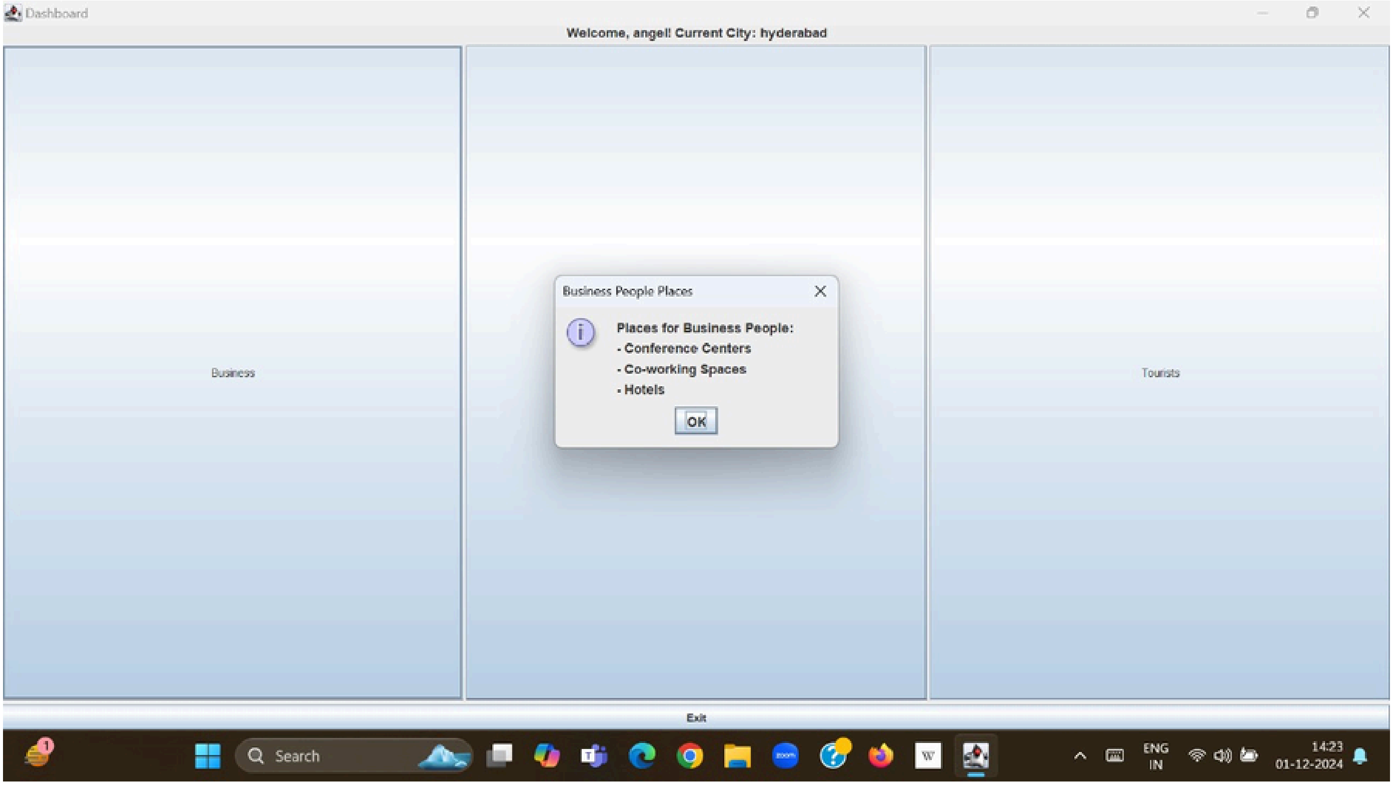
Name: angel

Email: a1234@gmail.com

City: hyderabad



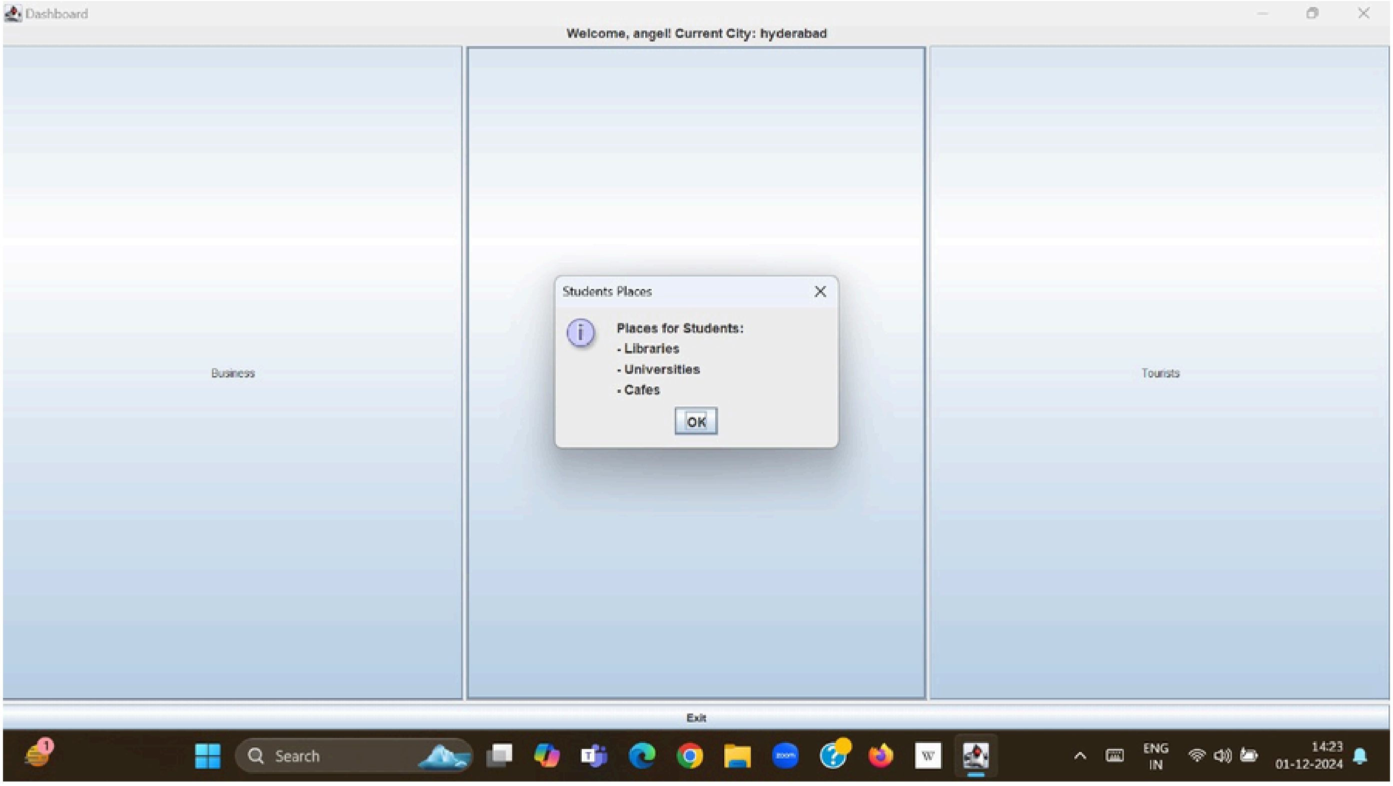
Welcome, angel! Current City: hyderabad



Business People Places

Places for Business People:
- Conference Centers
- Co-working Spaces
- Hotels

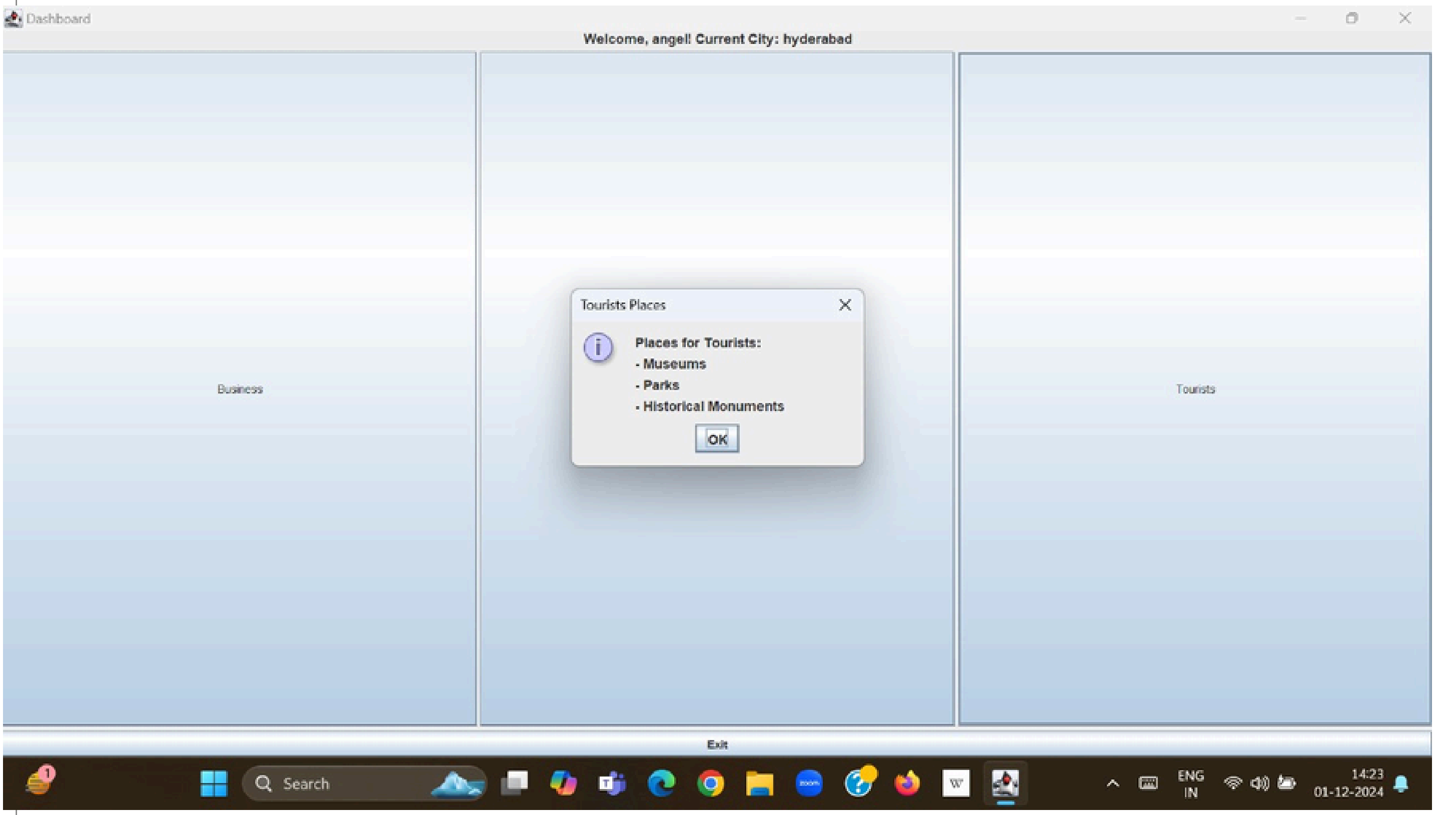
OK



Students Places

Places for Students:
- Libraries
- Universities
- Cafes

OK



Tourists Places

Places for Tourists:
- Museums
- Parks
- Historical Monuments

OK

CONCLUSION:

The Smart City Project is a comprehensive and practical web-based application that serves as a digital city guide. It not only benefits tourists by providing detailed insights into a city's attractions, transportation, and accommodations but also aids residents and businesses by acting as a central information hub. With its modular design and user-friendly interface, this project demonstrates the potential of technology to enhance urban living and tourism experiences.

By integrating future enhancements such as real-time updates, AI-based personalization, and mobile accessibility, the application can evolve into a robust platform capable of meeting the needs of diverse users. Overall, the Smart City Project exemplifies how innovative software solutions can bridge the information gap and contribute to a smarter, more connected urban environment.

THE END