



A Project Report

on

CITY GUIDE PROJECT

Submitted in partial fulfillment of requirements for the award of the course

of

EGB1201-JAVA PROGRAMMING

Under the guidance of

Mrs.N.SRIJA M.E.(Ph.D.),

Assistant Professor / IT

Submitted By

SANDHIYA R

(927624BEC182)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

M.KUMARASAMY COLLEGE OF ENGINEERING

(Autonomous)

KARUR – 639 113

DECEMBER 2025

M. KUMARASAMY COLLEGE OF ENGINEERING

(Autonomous Institution affiliated to Anna University, Chennai)

KARUR – 639 113

BONAFIDE CERTIFICATE

This is to certify that this project report on “**CITY GUIDE PROJECT**” is the bonafide work of **SANDHIYA R (927624BEC182)** who carried out the project work during the academic year 2025 - 2026 under my supervision.

Signature

Mrs.N.SRIJA M.E.(Ph.D.),

SUPERVISOR,

Department of Information Technology,

M.Kumarasamy College of Engineering,

Thalavapalayam, Karur -639 113.

Signature

Dr.N. MAHENDRAN, Ph.D.,

HEAD OF THE DEPARTMENT,

Department of Electronics and
Communication Engineering,

M.Kumarasamy College of Engineering,

Thalavapalayam, Karur -639 113.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VISION OF THE INSTITUTION

To emerge as a leader among the top institutions in the field of technical education

MISSION OF THE INSTITUTION

- Produce smart technocrats with empirical knowledge who can surmount the global challenges
- Create a diverse, fully-engaged, learner-centric campus environment to provide quality education to the students
- Maintain mutually beneficial partnerships with our alumni, industry, and Professional associations

DEPARTMENT VISION, MISSION, PEO, PO AND PSO

VISION

To empower the Electronics and Communication Engineering students with emerging technologies, professionalism, innovative research and social responsibility.

MISSION

- M1:** Attain the academic excellence through innovative teaching learning process, research areas & laboratories and Consultancy projects.
- M2:** Inculcate the students in problem solving and lifelong learning ability.
- M3:** Provide entrepreneurial skills and leadership qualities.
- M4:** Render the technical knowledge and skills of faculty members.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

- **PEO1:** Core Competence: Graduates will have a successful career in academia or industry associated with Electronics and Communication Engineering
- **PEO2:** Professionalism: Graduates will provide feasible solutions for the challenging problems through comprehensive research and innovation in the allied areas of Electronics and Communication Engineering.
- **PEO3:** Lifelong Learning: Graduates will contribute to the social needs through lifelong learning, practicing professional ethics and leadership quality.

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- 10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- 11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- 12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- PSO1: Applying knowledge in various areas, like Electronics, Communications, Signal processing, VLSI, Embedded systems etc., in the design and implementation of Engineering application.
- PSO2: Able to solve complex problems in Electronics and Communication Engineering with analytical and managerial skills either independently or in team using latest hardware and software tools to fulfil the industrial expectations.

ACKNOWLEDGEMENT

We gratefully remember our beloved **Founder Chairman, (Late) Thiru. M. Kumarasamy**, whose vision and legacy laid the foundation for our education and inspired us to successfully complete this project.

We extend our sincere thanks to **Dr. K. Ramakrishnan, Chairman**, and **Mr. K. R. Charun Kumar, Joint Secretary**, for providing excellent infrastructure and continuous support throughout our academic journey.

We are privileged to extend our heartfelt thanks to our respected Principal, **B. S. Murugan, B.Tech., M.Tech., Ph.D.**, for providing us with a conducive environment and constant encouragement to pursue this project work.

We sincerely thank **Dr.N.Mahendran M.E.,Ph.D., Professor and Head, Department of Electronics and Communication Engineering**, for his continuous support, valuable guidance, and motivation throughout the course of this project.

Our special thanks and deep sense of appreciation go to our **Project Supervisor, Mrs.N.SRIJA,M.E.(Ph.D.)**, Assistant Professor, **Department of Information Technology**, for her exceptional guidance, continuous supervision, constructive suggestions, and unwavering support, all of which have been instrumental in the successful execution of this project.

We would also like to acknowledge **Dr.G.REVATHI,M.E.,Ph.D., Assistant Professor, our Class Advisor** for his constant encouragement and coordination that contributed to the smooth progress and completion of our project work.

We gratefully thank all the **faculty members of the Department of Electronics and Communication Engineering** for their timely assistance, valuable insights, and constant support during various phases of the project.

Finally, we extend our profound gratitude to our **parents and friends** for their encouragement, moral support, and motivation, without which the successful completion of this project would not have been possible.

ABSTRACT

The **City Guide Project** is designed to help users easily explore and navigate a city by providing essential information about important locations, such as tourist attractions, restaurants, hospitals, educational institutions, shopping centers, and transportation facilities. The main objective of this project is to create a user-friendly platform that acts as a digital guide, offering quick access to location details, routes, and contact information.

This project integrates data management and user interface design to ensure accurate and efficient information retrieval. Users can search for a specific place or browse categories to find recommendations based on their needs.

The system is particularly useful for tourists and new residents who are unfamiliar with the city. Overall, it enhances the travel experience by providing reliable city information in an organized and accessible format, reducing the need for manual searches and improving convenience through digital automation.

ABSTRACT WITH POs AND PSOs MAPPING

ABSTRACT	COs MAPPED	POs MAPPED	PSOs MAPPED
The City Guide Project helps users explore and find important places in a city such as tourist spots, hotels, and hospitals. It provides location details and routes in a simple digital format. This project makes city navigation easier and more convenient for users and visitors.	CO1	PO1	PSO1
	CO2	PO2	PSO2
	CO3	PO3	
	CO4	PO4	
	CO5	PO5	
		PO6	
		PO7	
		PO8	
		PO9	
		PO10	
		PO11	
		PO12	

Note: 1- Low , 2-Medium , 3- High

SUPERVISOR

HEAD OF THE DEPARTMENT

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	VII
1	INTRODUCTION	1
	1.1 OBJECTIVE	1
	1.2 OOPS CONCEPTS	1
	1.3 LITERATURE SURVEY	2
2	PROJECT METHODOLOGY	3
	2.1 PROPOSED WORK	3
	2.2 ARCHITECTURE	4,5
3	MODULE DESCRIPTION	6
	3.1 USER MODULE	6
	3.2 INFORMATION MODULE	6
	3.3 SEARCH MODULE	6
	3.4 TRANSPORT MODULE	6



	3.5 EMERGENCY MODULE	6
	3.6 ADMIN MODULE	7
4	RESULTS AND DISCUSSION	8,9
5	CONCLUSION	10
	REFERENCES	11
	APPENDIX	12-15

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

1. To provide detailed information about important places in the city.
2. To help users easily locate routes and destinations.
3. To offer a user-friendly and interactive platform for city exploration.
4. To assist tourists and new residents in finding essential services quickly.
5. To save time by providing instant access to city information.

1.2 OOPS CONCEPTS

1. Class and Object:

Classes are like templates (for example, **Place**, **User**, **Location**).

Example: A Place class is created, and objects like Hotel, Park, and Temple are made from it.

2. Encapsulation:

Data is kept safe inside the class and can be changed only using methods.

Example: Place details like name and address are private and can be updated only using setName() or setAddress() methods.

3. Inheritance:

One class can use the features of another class.

Example: Restaurant and Hospital classes use the common features (like name and address) from the Place class.

4. Polymorphism:

One function name can have many forms.

Example: showDetails() can work differently for each place (Hotel, Park, Hospital).

5. Abstraction:

Shows only important details and hides the complex code from the user.

Example: The user searches a place without seeing how the search works inside the program.

1.3 LITERATURE SURVEY

1. Many existing city guide apps help users find nearby hotels, restaurants, and tourist spots easily.

Most of them use GPS and Google Maps to show locations and routes.

However, they often need continuous internet access to work properly.

2. Web-based city guide systems allow users to search and view places through online portals.

They display information on interactive maps and provide filters for categories.

But they can be complex for beginners and require registration in some cases.

3. Some research works focus on adding recommendation features in city guides.

These systems suggest popular or highly rated places based on user preferences.

However, they need large databases and user feedback to function effectively.

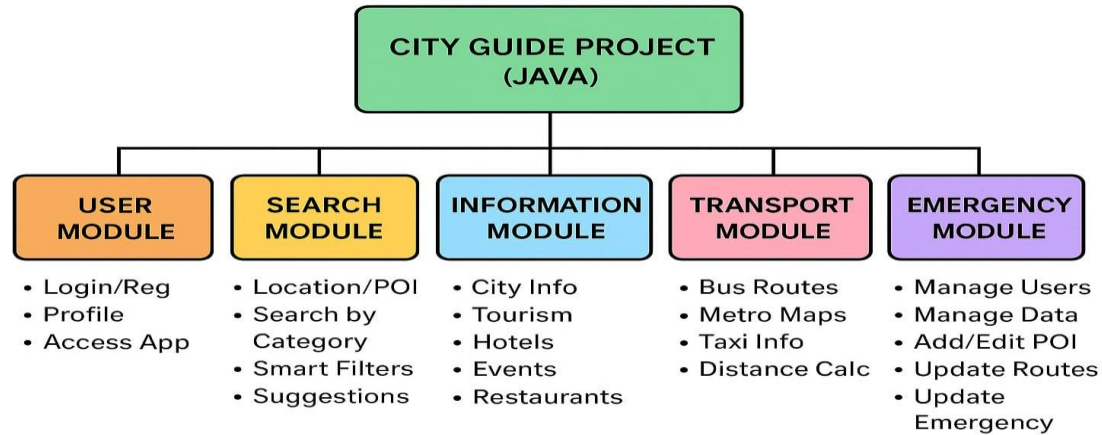
CHAPTER 2

PROJECT METHODOLOGY

2.1 PROPOSED WORK

1. The proposed City Guide System using Java is designed to help users easily find important places in a city such as hotels, restaurants, hospitals, tourist spots, and shopping centers. The system allows users to search for a place, view its details, and get route information.
2. The project uses Java for the main program and MySQL for storing data such as place names, addresses, categories, and contact details. The system will have two main users — Admin and User.
3. Admin: Can add, update, or delete place information and manage the city database.
4. User: Can search for places, view details, and get directions based on their preferences.
5. The system aims to make city exploration easier by providing accurate information quickly. It reduces manual searching, saves time, and offers a simple interface for both locals and tourists.

2.2 ARCHITECTURE



1. User Layer:

- Allows users to search for places, view details, and get directions.
- Provides a simple and interactive interface for easy navigation.

2. Information Layer:

- Stores all city data like place names, addresses, and contact details in the database.
- Ensures accurate and updated information is available for users.

3. Search Layer:

- Processes user queries to find matching places quickly.
- Filters and displays results based on keywords or selected categories.

4. Transport Layer:

- Helps users find routes and nearby transport options.
- Connects with maps to show directions and travel distance.

5. Emergency Layer:

- Provides quick access to hospitals, police stations, and fire services.
- Shows the nearest emergency locations with contact details.

6. Admin Layer:

- Allows the admin to add, edit, or delete place information.
- Manages user data and ensures the system remains updated and secure.

CHAPTER 3

MODULE DESCRIPTION

1. User Module:

- Allows users to search for places such as hotels, hospitals, restaurants, and tourist spots.
- Displays detailed information like address, contact number, and route to the selected place.

2. Admin Module:

- Admin can add, update, or delete place details in the database.
- Manages city information and ensures all data shown to users is accurate.

3. Search Module:

- Handles user queries to find places quickly based on category or keyword.
- Filters results and displays only the most relevant locations.

4. Information Module:

- Stores and retrieves city data such as place names, types, and locations from the database.
- Ensures secure access and smooth data management between user and admin.

5. Transport Module:

- Provides map-based route details to reach the selected destination.
- Helps users find nearby bus stops, railway stations, or transport options.

6. Emergency Module:

- Displays nearby emergency services like hospitals, police stations, and fire stations.
- Provides quick contact and route details for emergency help.

CHAPTER 4

RESULTS AND DISCUSSION



Fig.4.1

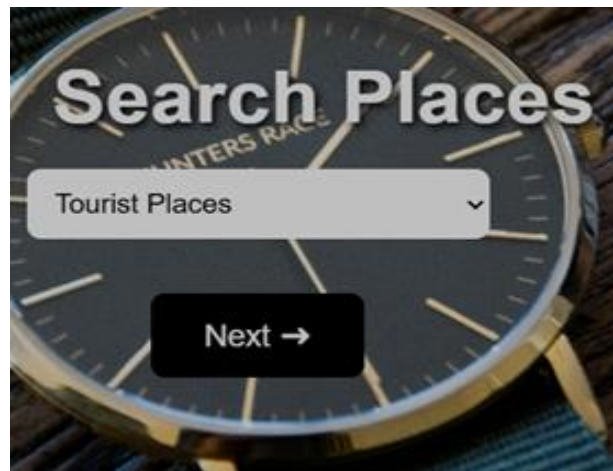


Fig.4.2

Register

sandhiya

sandhiya@gmail.com

.....

Register

Fig.4.3

Transport Info

- Bus routes available
- Auto/Taxi stands
- Metro connectivity details

Next →

Fig.4.4



Emergency Services

- Police: 100
- Ambulance: 108
- Fire Service: 101

Back to Home

Fig.4.5

CHAPTER 5

CONCLUSION

The **City Guide System** helps users easily find important places in the city such as hotels, hospitals, restaurants, shopping centers, and tourist attractions. It provides complete details like address, contact number, and route information, making city navigation faster and more convenient. The system reduces manual searching and saves time by offering accurate and real-time information. The admin can easily manage data by adding, updating, or deleting place details whenever required, ensuring that users always receive the latest information.

The simple and interactive interface makes the system easy to use for both local residents and tourists. It also includes special modules for transport and emergency services, allowing users to reach their destinations safely and quickly. By integrating Java and MySQL, the system ensures efficiency, reliability, and smooth data management. Overall, the project successfully meets its objectives by providing a smart, digital, and user-friendly solution for exploring and navigating a city. In the future, additional features like GPS tracking, user reviews, and offline access can be added to make the system even more powerful and useful.

REFERENCES:

- E. Balagurusamy, *Programming with Java – A Primer*, Tata McGraw-Hill, 6th Edition, 2017.
- Cay S. Horstmann, *Core Java Volume I – Fundamentals*, Pearson Education, 11th Edition, 2022.
- MySQL Documentation – <https://dev.mysql.com/doc/>
- Oracle Java Tutorials – <https://docs.oracle.com/javase/tutorial/>
- Research papers and articles on City Information Systems, Location-Based Services, and Online Mapping Applications from IEEE Xplore and other open-source platforms.

APPENDIX

(Coding)

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

public class CityGuide extends JFrame {

    CardLayout cardLayout;
    JPanel mainPanel;

    public CityGuide() {


        // Window setup
        setTitle("City Guide Portal");
        setSize(900, 600);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        cardLayout = new CardLayout();
        mainPanel = new JPanel(cardLayout);

        // Add screens
        mainPanel.add(homeScreen(), "home");
        mainPanel.add(loginScreen(), "login");
        mainPanel.add(registerScreen(), "register");
        mainPanel.add(searchScreen(), "search");
        mainPanel.add(transportScreen(), "transport");
        mainPanel.add(emergencyScreen(), "emergency");

        add(mainPanel);
        setVisible(true);
    }

    // ----- HOME SCREEN -----
    JPanel homeScreen() {
        JPanel p = new JPanel();
        p.setLayout(new GridBagLayout());
        p.setBackground(new Color(50, 130, 200));

        JLabel title = new JLabel(" City Guide Portal");
        title.setFont(new Font("Poppins", Font.BOLD, 36));
        title.setForeground(Color.WHITE);

        JLabel sub = new JLabel("<html><center>Discover your city with simple navigation —<br>Search  
places, transport routes, and emergency services.</center></html>");
        sub.setForeground(Color.WHITE);
        sub.setFont(new Font("Arial", Font.PLAIN, 18));

        JButton next = new JButton("Next →");
        next.setFont(new Font("Arial", Font.BOLD, 18));
```

```

next.addActionListener(e -> cardLayout.show(mainPanel, "login"));

GridBagConstraints c = new GridBagConstraints();
c.insets = new Insets(20, 20, 20, 20);

c.gridy = 0; p.add(title, c);
c.gridy = 1; p.add(sub, c);
c.gridy = 2; p.add(next, c);

return p;
}

// ----- LOGIN SCREEN -----
JPanel loginScreen() {
    JPanel p = new JPanel();
    p.setLayout(new GridBagLayout());
    p.setBackground(new Color(40, 40, 40));

    JLabel title = new JLabel("🔒 Login");
    title.setFont(new Font("Poppins", Font.BOLD, 32));
    title.setForeground(Color.WHITE);

    JTextField user = new JTextField(15);
    user.setFont(new Font("Arial", Font.PLAIN, 16));

    JPasswordField pass = new JPasswordField(15);

    JButton login = new JButton("Login");
    login.addActionListener(e -> cardLayout.show(mainPanel, "register"));

    GridBagConstraints c = new GridBagConstraints();
    c.insets = new Insets(15, 15, 15, 15);

    c.gridy = 0; p.add(title, c);
    c.gridy = 1; p.add(user, c);
    c.gridy = 2; p.add(pass, c);
    c.gridy = 3; p.add(login, c);

    return p;
}

// ----- REGISTER SCREEN -----
JPanel registerScreen() {
    JPanel p = new JPanel();
    p.setLayout(new GridBagLayout());
    p.setBackground(new Color(30, 30, 30));

    JLabel title = new JLabel("📝 Register");
    title.setFont(new Font("Poppins", Font.BOLD, 32));
    title.setForeground(Color.WHITE);

    JTextField name = new JTextField(15);
    JTextField email = new JTextField(15);

```

```

JPasswordField pass = new JPasswordField(15);

JButton next = new JButton("Register");
next.addActionListener(e -> cardLayout.show(mainPanel, "search"));

GridBagConstraints c = new GridBagConstraints();
c.insets = new Insets(15, 15, 15, 15);

c.gridy = 0; p.add(title, c);
c.gridy = 1; p.add(name, c);
c.gridy = 2; p.add(email, c);
c.gridy = 3; p.add(pass, c);
c.gridy = 4; p.add(next, c);

return p;
}

// ----- SEARCH SCREEN -----
JPanel searchScreen() {
    JPanel p = new JPanel();
    p.setLayout(new GridBagLayout());
    p.setBackground(new Color(70, 130, 180));

    JLabel title = new JLabel("🔍 Search Places");
    title.setFont(new Font("Poppins", Font.BOLD, 32));
    title.setForeground(Color.WHITE);

    String[] places = {"Malls", "Tourist Places", "Restaurants", "Hospitals"};
    JComboBox<String> box = new JComboBox<>(places);

    JButton next = new JButton("Next ➡");
    next.addActionListener(e -> cardLayout.show(mainPanel, "transport"));

    GridBagConstraints c = new GridBagConstraints();
    c.insets = new Insets(15, 15, 15, 15);

    c.gridy = 0; p.add(title, c);
    c.gridy = 1; p.add(box, c);
    c.gridy = 2; p.add(next, c);

    return p;
}

// ----- TRANSPORT SCREEN -----
JPanel transportScreen() {
    JPanel p = new JPanel();
    p.setLayout(new GridBagLayout());
    p.setBackground(new Color(0, 40, 80));

    JLabel title = new JLabel("🚗 Transport Info");
    title.setFont(new Font("Poppins", Font.BOLD, 32));
    title.setForeground(Color.WHITE);

```



```

JLabel info = new JLabel("<html>• Bus routes available<br>• Auto/Taxi stands<br>• Metro connectivity
details</html>");
info.setFont(new Font("Arial", Font.PLAIN, 18));
info.setForeground(Color.WHITE);

JButton next = new JButton("Next →");
next.addActionListener(e -> cardLayout.show(mainPanel, "emergency"));

GridBagConstraints c = new GridBagConstraints();
c.insets = new Insets(15, 15, 15, 15);

c.gridy = 0; p.add(title, c);
c.gridy = 1; p.add(info, c);
c.gridy = 2; p.add(next, c);

return p;
}

// ----- EMERGENCY SCREEN -----
JPanel emergencyScreen() {
    JPanel p = new JPanel();
    p.setLayout(new GridBagLayout());
    p.setBackground(Color.BLACK);

    JLabel title = new JLabel("🚒 Emergency Services");
    title.setFont(new Font("Poppins", Font.BOLD, 32));
    title.setForeground(Color.WHITE);

    JLabel info = new JLabel("<html>• Police: 100<br>• Ambulance: 108<br>• Fire Service: 101</html>");
    info.setFont(new Font("Arial", Font.PLAIN, 20));
    info.setForeground(Color.WHITE);

    JButton back = new JButton("Back to Home");
    back.addActionListener(e -> cardLayout.show(mainPanel, "home"));

    GridBagConstraints c = new GridBagConstraints();
    c.insets = new Insets(15, 15, 15, 15);

    c.gridy = 0; p.add(title, c);
    c.gridy = 1; p.add(info, c);
    c.gridy = 2; p.add(back, c);

    return p;
}

// ----- MAIN -----
public static void main(String[] args) {
    new CityGuide();
}
}

```