

Smart Contextual Information Device

A

PBL-II Project Report submitted to Savitribai Phule Pune University, Pune



In partial Fulfillment for the awards of Degree of Engineering in
Computer Engineering

Submitted by

Ms. Renuka Dawane(COA223)

Ms. Payal Dhumal(COA229)

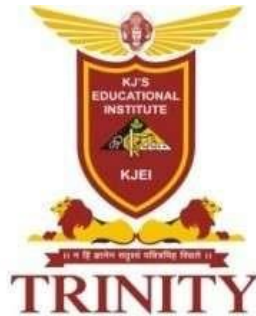
Ms. Vaishanvi Dudhane(COA231)

Ms. Nikita Girbane (COA238)

Ms. Pratiksha Gorde (COA240)

Under the Guidance of

Prof. Shaikh Jareena



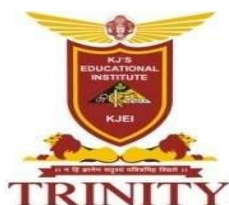
A.Y. 2024-25

Department of Computer Engineering
Trinity College of Engineering & Research, Pune



KJEI's
Trinity College of Engineering & Research, Pune
Department of Computer Engineering
(2024-25)

Certificate



This is to certify that,

Ms. Renuka Dawane(COA223)
Ms. Payal Dhumal(COA229)
Ms. Vaishanvi Dudhane(COA231)
Ms. Nikita Girbane (COA238)
Ms. Pratiksha Gorde (COA240)

have successfully completed the PBL-II project entitled “Smart Contextual Information Device” in partial fulfillment of the requirements for the Second Year of Computer Engineering under the Savitribai Phule Pune University during the academic year 20242025

Date : 18-04-2025

Place:Pune

Prof.Shaikh Jareena
Project Guide

Prof. Shaikh Jareena
PBL Coordinator

Dr. Geetika Narang
Head, Department of Computer Engineering

Acknowledgements

With deep sense of gratitude we would like to thank all the people who have lit our path with their kind guidance. We are very grateful to these intellectuals who did their best to help during our project work.

It is our proud privilege to express a deep sense of gratitude to Dr. A. B. Auti, Principal of TCOER, Pune for his support to complete this project. We remain indebted to Dr. Geetika Narang, H.O.D. Computer Engineering Department for his timely suggestion and valuable guidance.

The special gratitude goes to Prof. Shaikh Jareena, PBL Coordinator for her excellent and precious guidance in completion of this work. We would like to express sincere thanks to Prof. Shaikh Jareena for her constant support and valuable guidance for completion of this project. We thanks to all the department faculty members for their appreciable help for our working project.

We are also thankful to our parents who provided their wishful support for our project completion successfully. And lastly we thank our all friends and the people who are directly or indirectly related to our project work.

Ms. Renuka Dawane(COA223)

Ms. Payal Dhumal(COA229)

Ms. Vaishanvi Dudhane(COA231)

Ms. Nikita Girbane (COA238)

Ms. Pratiksha Gorde (COA240)

VISION & MISSION OF THE INSTITUTE

INSTITUTE VISION

To be premier knowledge center for socio-economic development.

INSTITUTE MISSION

To provide education that combines rigorous academics with joy of discovery through sustained efforts and dynamic strategies in building innovative, participatory, problem based learning practices and research that leads to capacity building of students.

VISION & MISSION OF THE COMPUTER ENGINEERING DEPARTMENT

DEPARTMENT VISION

To be a Centre of Excellence that adapts to the changing technological scenario in the field of Computer Engineering for socio-economic development

DEPARTMENT MISSION

M1: To provide learning ambience with rigorous academics and sustained efforts.

M2: To develop technically competent professionals with dynamic strategies in building innovative, participatory, problem based learning practices. M3: To inculcate professionals with social and ethical values that leads to capacity building

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO 1: To produce graduates who have strong foundation of knowledge and skills in field of computer engineering.

PEO 2: To apply current tools and techniques to create systems for solving industry oriented problems.

PEO 3: To develop the ability of innovative thinking, and decision making skills for solving real life problems.

PEO4: To develop life-long learning, professional and ethical attitude making positive impact on society.

PROGRAM SPECIFIC OBJECTIVES (PSOs)

Graduates will be able:

PSO 1: To apply computer engineering concepts with appropriate tools and technologies for design and development of solutions for real time problems.

PSO 2: To cultivate ethical skills and cater the needs of the industry and society with principles on engineering and core human values.

Abstract

The **Easy Java Guide** is an educational desktop application developed using Java Swing that enables users to easily search, explore, and understand fundamental Java programming concepts. Designed with a modern, user-friendly graphical interface, the application allows learners to interactively browse Java topics such as OOP principles, multithreading, exception handling, and more. Each topic includes clear definitions, practical syntax examples, and usage explanations, making it suitable for beginners and intermediate learners.

Key features include an intelligent search field with real-time suggestions, animated text display for results (typing effect), and intuitive navigation through Start, Back, and Exit buttons. The application leverages object-oriented programming principles and GUI design to enhance the learning experience and promote efficient knowledge acquisition in Java programming.

Index

Sr.no	Topic	Page No
1	Chapter 1 (Introduction)	10
2	Chapter 2 (Literature Survey)	11
3	Chapter 3 (Problem Statement)	13
4	Chapter 4 (Software Requirement Specification)	14
5	Chapter 5 (System Design)	15
6	Chapter 6 (Result/Outcome)	18
7	Chapter 7 (Conclusion & Future Scope)	21
8	References	22

Figure Index

Sr.no	Topic	Page No
1	Block Diagram- Easy Java Guide	15
2	First Page: Welcome To Java Topic Search	18
3	Second Page: Search Page 1. Second Page: Topic Search 2. Second page: Searched Topic-Encapsulation 3. Second Page: Searched Topic - Polymorphism	19
4	Final Page: Thank You	20

Chapter 1

Introduction

1.1 Overview

Easy Java Guide is a Java-based desktop application designed to assist users in learning and exploring Java programming concepts through an interactive graphical user interface. Built using Java Swing, the application provides a searchable database of key Java topics such as data types, OOP concepts, exception handling, multithreading, and more. With real-time suggestions and a clean, animated interface, it helps users quickly access definitions, syntax, and practical usage examples, making it a valuable tool for students and beginners.

1.2 Aim

The aim of this project is to develop an educational application that simplifies the process of learning Java programming by offering instant, structured, and visually appealing access to key concepts through a user-friendly GUI.

1.3 Objective

- To create a Java Swing-based application that allows topic-wise searching of Java concepts.
-
- To provide clear definitions, syntax, and usage of Java topics in a readable format.
-
- To implement real-time suggestions to enhance search efficiency.
-
- To make the learning process engaging with animations and intuitive navigation.
-
- To promote better understanding and retention of Java programming fundamentals for beginners.

Chapter 2

Literature Survey

Learning programming languages like Java has traditionally relied on textbooks, online tutorials, and coding platforms. However, many beginner learners face challenges in locating precise, example-driven explanations of programming concepts in a single, interactive environment. Most existing educational tools are either overly complex for novices or lack engaging interfaces, leading to reduced motivation and slower learning. Desktop-based educational applications using Java Swing offer a valuable middle ground by combining interactivity with simplicity and offline access. The Easy Java Guide project addresses this gap by offering an easy-to-use, topic-searchable interface that presents key Java concepts with definitions, syntax, and use cases.

Existing Systems

1. Online Learning Platforms (e.g., GeeksforGeeks, W3Schools, JavaPoint)

These websites offer a wide range of tutorials, syntax references, and code examples for Java and other programming languages. They are accessible from any internet-connected device.

Advantages:

- Comprehensive Content: Covers a broad spectrum of Java topics from basics to advanced.
- Free Access: Many platforms offer open-access resources.
- Interactive Editors: Some platforms provide online code compilers for practice.

Disadvantages:

- Internet Dependency: Requires a stable internet connection for access.
- Distraction Risks: Browsing multiple tabs or ads may reduce focus.
- Overwhelming Layouts: Beginners may find the layout and content density hard to navigate..

2. Java IDEs with Documentation (e.g., Eclipse, IntelliJ IDEA)

Integrated Development Environments (IDEs) support Java development and offer inline documentation and syntax suggestions.

Advantages:

- Built-in Help: Provides Java documentation and syntax suggestions while coding.

- **Efficient Coding:** Enables real-time error checking and debugging.

Disadvantages:

- **Complexity for Beginners:** IDEs have steep learning curves.
- **Resource Intensive:** Consumes significant system resources.
- **Not Focused on Theory:** More geared towards development than conceptual understanding.

3. E-Learning Apps (e.g., SoloLearn, Programming Hub)

Mobile-based educational apps designed for coding practice and learning.

Advantages:

- **Mobile Friendly:** Learn on the go.
- **Gamified Learning:** Offers quizzes and achievements to keep users engaged.
- **Beginner-Oriented:** Simplified content for new learners.

Disadvantages:

- **Limited Depth:** Often lacks in-depth explanations or real-world coding context.
- **No Desktop Integration:** Not ideal for learners who prefer typing and practicing on PC.
- **Minimal Customization:** Restricted learning paths and fixed lesson orders.

4. Easy Java Guide (Proposed System)

A desktop-based Java learning application built with Java Swing that allows users to search for Java topics and view definitions, syntax, and uses in an engaging, animated GUI.

Advantages:

- **Offline Access:** Works without the internet, making it reliable anytime.
- **Interactive UI:** Real-time suggestions and animated text output improve user experience.
- **Focused Content:** Designed specifically to explain Java concepts clearly for learners.

Disadvantages:

- **Limited Scope:** Currently focuses only on Java and not other languages or tools.
- **Manual Updates:** Content additions or corrections require manual changes in code.
- **Desktop Only:** Not available on mobile or web platforms.

Chapter 3

Problem Statement

Despite the abundance of online resources for learning Java, beginners often struggle to find a centralized, easy-to-navigate platform that presents programming concepts with clarity, structure, and simplicity. Existing solutions like IDEs, websites, and mobile apps either overwhelm users with complex interfaces or offer shallow, non-interactive learning experiences. Additionally, most educational platforms require constant internet access and are not tailored specifically for beginners.

Chapter 4

Software Requirements Specification

1. Software Requirements

- Language: Java
- Framework: Java Swing (GUI components)
- Development Environment: Any Java IDE (e.g., IntelliJ IDEA, Eclipse, NetBeans)
- Java Version: JDK 8 or higher
- Operating System: Windows, macOS, or Linux (Java-compatible OS)

2. Hardware Requirements

- Processor: Intel Core i3 or equivalent
- RAM: Minimum 2 GB
- Storage: Minimum 100 MB available space
- Display: Minimum 1024x768 resolution

3. Functional Requirements

- User Interface
- Display a welcome screen with a "Start Searching" button.
- Provide a search field with auto-suggestions as the user types.
- Show results in a text area with animated (typing effect) output.
- Include navigation buttons: Back and Exit.

3.1 Search Functionality

- Allow users to search for predefined Java topics.
- Display the topic's definition, syntax, and use case.

3.2 Suggestions

- Generate real-time suggestions based on partial input in the search field.

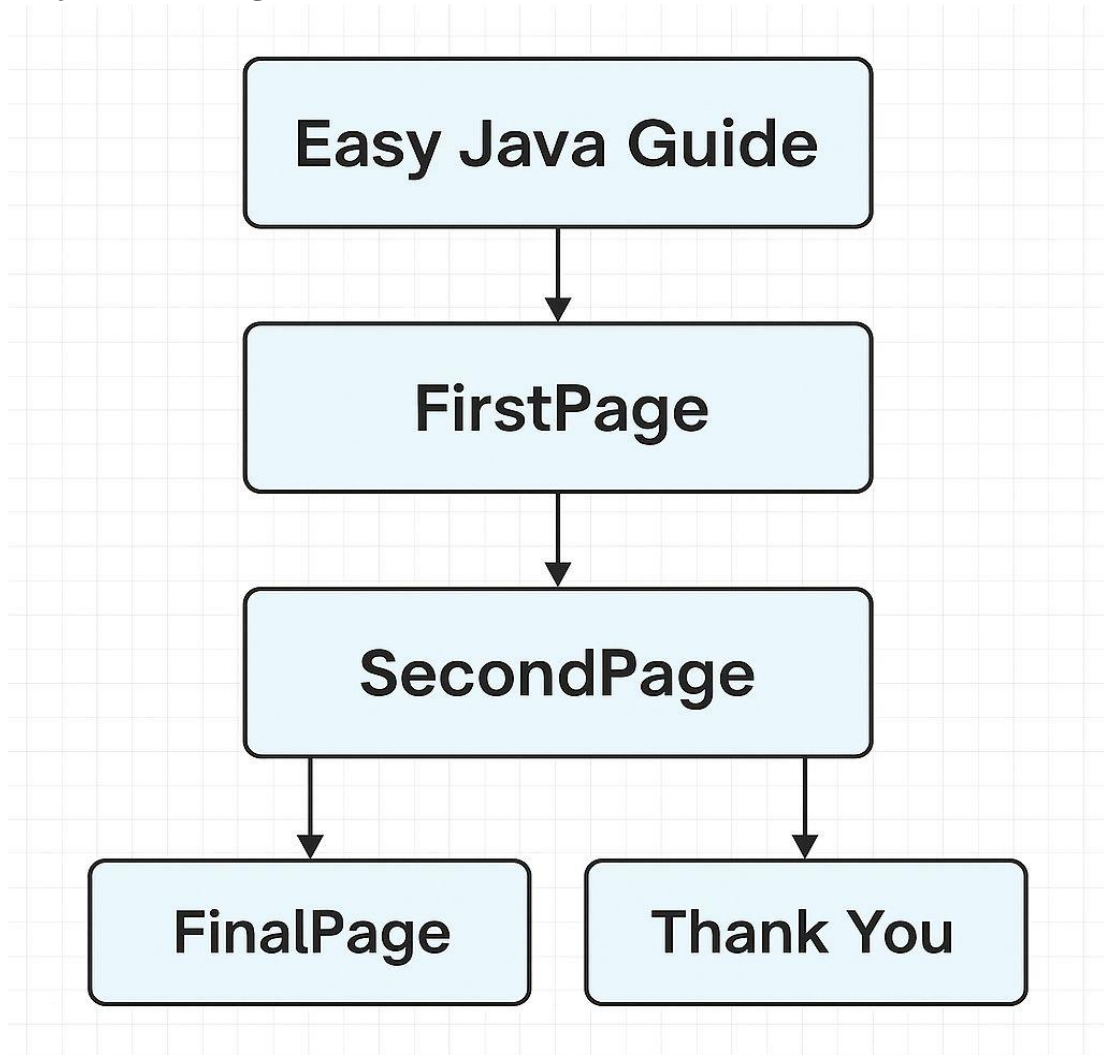
3.4 Navigation

- Transition between screens: Welcome → Search Page → Thank You page.

Chapter 5

System Design

5.1 Project Block Diagram



1.Block Diagram

5.2Methodology Used

The methodology for the Easy Java Guide involves several structure

steps:

1. System Initialization

- Load and initialize GUI components using Java Swing.
- Create a HashMap to store Java topics as keys and their respective detailed information (definition, syntax, and usage) as values.
- Load predefined topics such as Java Basics, OOP Concepts, Exception Handling, etc.
- Store all topic titles in a List to enable auto-suggestions during user input.
- Prepare the layout for the welcome screen, search page, and final thank-you screen.
- Utilize Java Swing to build the graphical user interface.
- Set up the main window structure, including:
 - Welcome Screen (FirstPage)
 - Topic Search Screen (SecondPage)
 - Final Thank You Screen (FinalPage)
- Apply consistent styling using color themes, fonts, and layout managers to ensure readability and aesthetic appeal

2. Topic Loading

- Initialize a HashMap with Java topics as keys and detailed content (definition, syntax, and usage) as values.
- Populate a list with all topic names for suggestion functionality.
- Create a HashMap to store Java topics as keys and their respective detailed information (definition, syntax, and usage) as values.
- Load predefined topics such as Java Basics, OOP Concepts, Exception Handling, etc.
- Store all topic titles in a List to enable auto-suggestions during user input.

3. Context Awareness (User Input Handling)

- Monitor user input in the search text field.
- Provide real-time suggestions by matching typed input with stored topics.
- Filter suggestions dynamically as the user types.
- Continuously monitor user input in the search field.
- Implement a real-time suggestion system using a JPopupMenu:
- Filter the topic list based on user keystrokes.
- Display dynamic dropdown suggestions that match the input.
- Update the suggestions as the user continues typing.

4. Decision-Making

When a user selects a topic or presses the "Search" button, check for a valid match in the topic database.

If found, retrieve and prepare the corresponding content.

5. Trigger Output

Display the topic's information with a typing animation for better engagement.

If no matching topic is found, display a “No matching topic found” message.

6. Navigation & Final Display

Allow users to go back to the welcome screen or exit the application.

On exit, display a final “Thank You” message with a minimal, centered layout.

7. Testing and Improvement

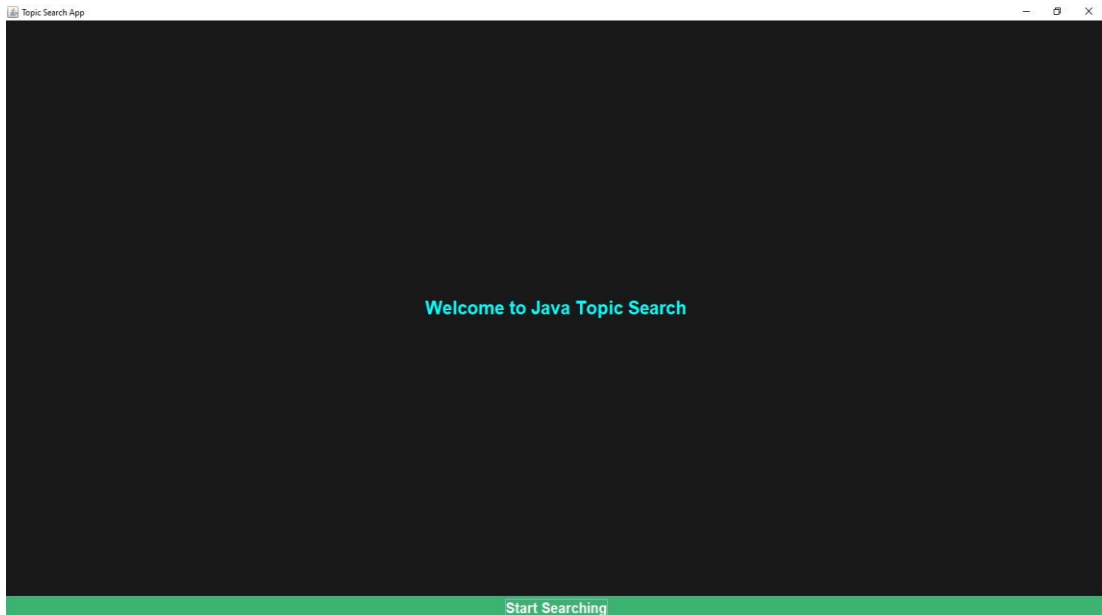
Test the application across different systems and screen sizes.

Adjust font sizes, colors, and layout for better accessibility and user experience.

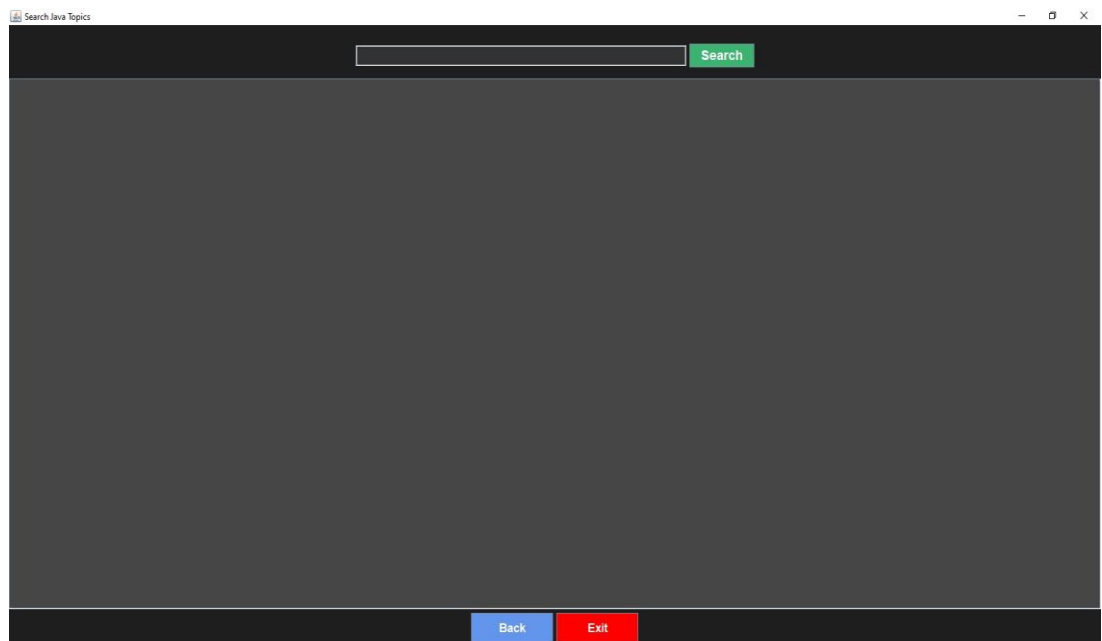
Debug search and display logic for smooth performance.

Chapter 6

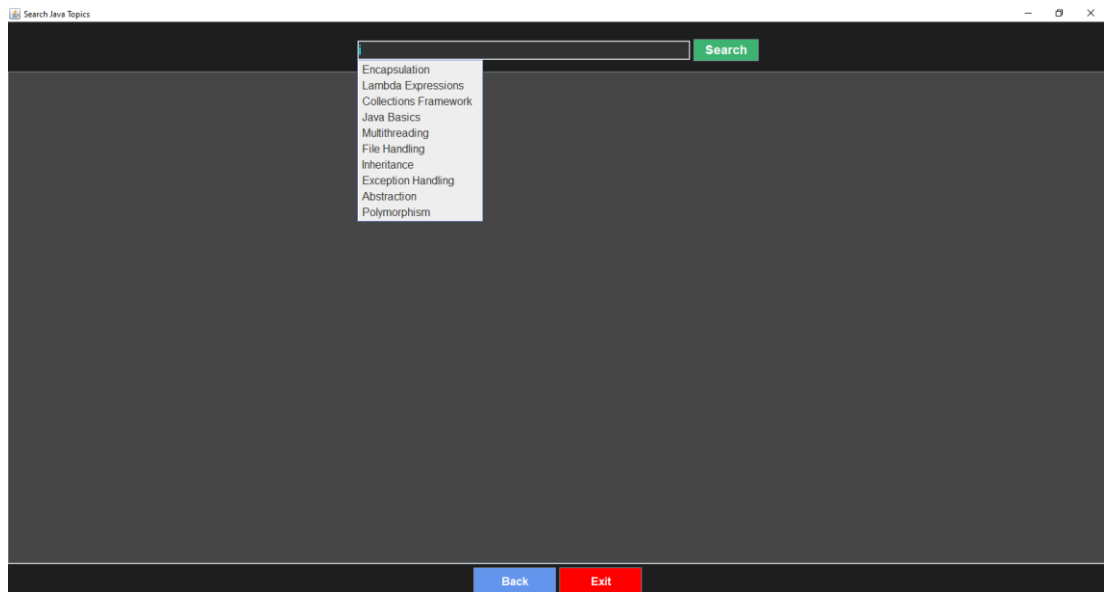
Result/Outcome



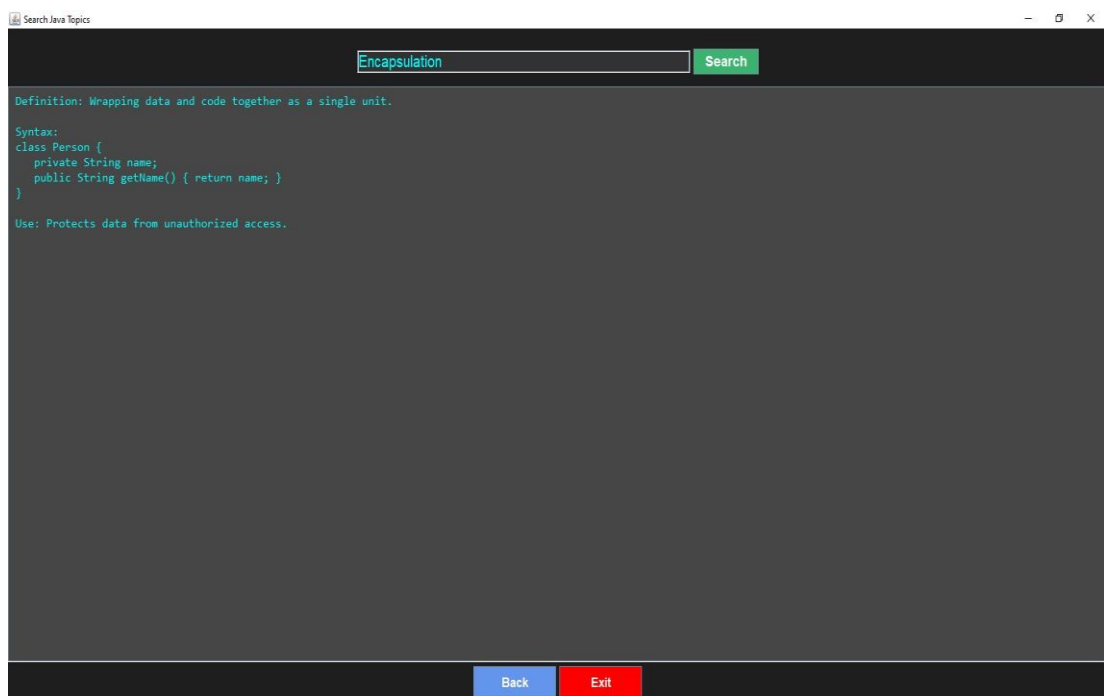
2. First Page: Welcome To Java Topic Search



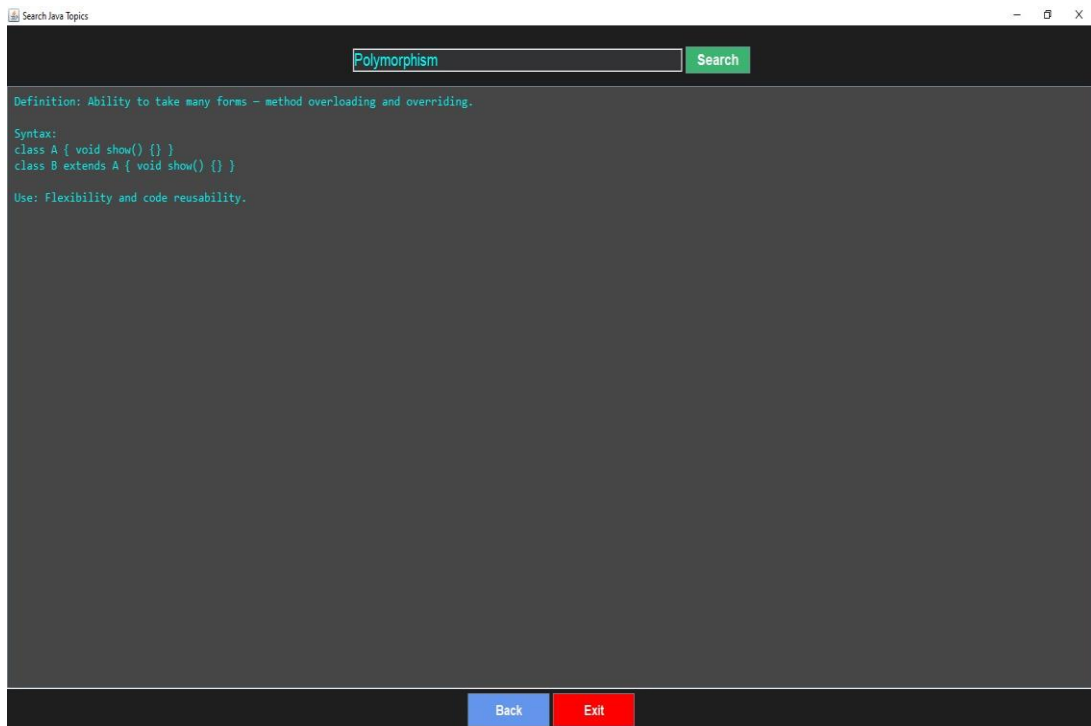
3. Second Page: Search Page



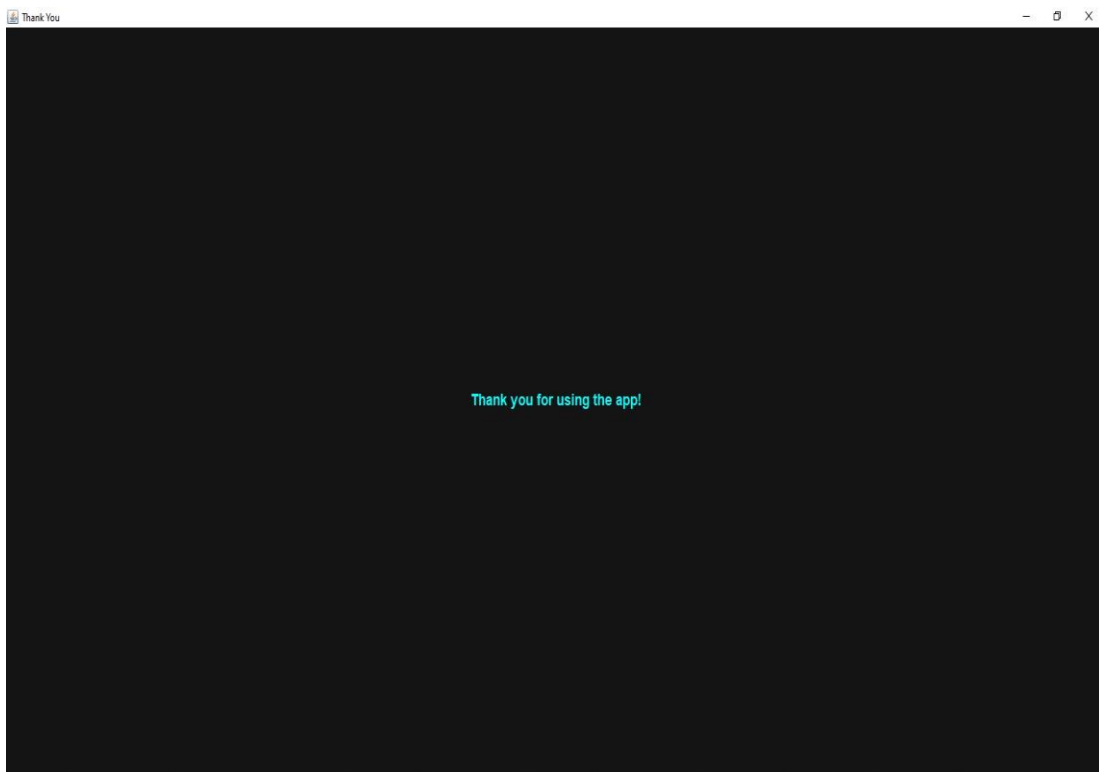
3.1 . Second Page: Topic Search



3.2 Second page: Searched Topic-Encapsulation



3.3 Second Page: Searched Topic - Polymorphism



4. Final Page: Thank You

Chapter 7

Conclusion

The **Easy Java Guide** project effectively provides a user-friendly platform for beginners to explore and learn fundamental Java programming concepts. With an intuitive interface, real-time search functionality, and dynamic topic displays, it simplifies the learning process. The system covers key Java topics like OOP, exception handling, and multithreading, making it an engaging tool for new programmers. With the potential for future enhancements, such as interactive quizzes, mobile integration, and code execution features, this project serves as a solid foundation for enhancing Java education in an accessible and interactive way.

Future Scope:

1. Expand Topic Coverage:

- Additional Java Concepts: More advanced topics such as JavaFX, Spring Framework, and JUnit testing can be added to make the guide more comprehensive.
- Integration with External Resources: Incorporating links to documentation, tutorials, or video courses would provide learners with additional learning material.

2. Interactive Quizzes:

- Introducing quizzes or mini-tests after each topic can help reinforce learning and assess the user's understanding. This would transform the application into an interactive study tool.

3. Mobile Application Version:

- A mobile version of the app could be developed, allowing users to learn on the go. This would expand accessibility and engagement, enabling users to access the guide on smartphones and tablets.

4. Dark/Light Theme Toggle:

- Adding a theme toggle feature would enhance accessibility, allowing users to switch between a dark mode and light mode, depending on their preference or time of day.

5. Code Execution Feature:

- Integrating a code editor within the app, where users can write and run Java code directly, would greatly enhance the interactivity of the platform. This feature could provide instant feedback on code snippets, helping users learn by doing.

6. Cloud Synchronization:

- Allowing users to sync their progress (saved searches, bookmarks, or quizzes) across multiple devices via cloud integration would enhance the continuity of the learning experience.

References

Books:

1. **"Head First Java" by Kathy Sierra and Bert Bates**
A great resource for beginners learning Java, with a hands-on approach to understanding core Java concepts.
2. **"Effective Java" by Joshua Bloch**
A widely recommended book for improving Java programming skills with advanced techniques, best practices, and tips.
3. **"Java: The Complete Reference" by Herbert Schildt**
This comprehensive guide provides a thorough understanding of Java programming concepts and API details.

Websites and Documentation:

1. **Oracle Java Documentation**
<https://docs.oracle.com/en/java/>
Official Java documentation from Oracle, which provides detailed information about the Java language, libraries, and tools.
2. **GeeksforGeeks - Java Programming Language**
<https://www.geeksforgeeks.org/java/>
A platform offering tutorials, explanations, and examples for a variety of Java programming topics.
3. **W3Schools - Java Tutorial**
<https://www.w3schools.com/java/>
An accessible online resource for learning Java, including basic syntax, OOP concepts, and more.
4. **Java Swing Documentation**
<https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>
Official documentation for **Swing**, the GUI toolkit used in the project to build the user interface.
5. **Stack Overflow**
<https://stackoverflow.com/questions/tagged/java>
A community-driven platform to ask specific Java-related questions and find answers from developers worldwide.

Tools and Libraries:

1. **Java Development Kit (JDK)**
<https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>
The official JDK for compiling and running Java applications.
2. **NetBeans IDE**
<https://netbeans.apache.org/>
An integrated development environment (IDE) for Java that can be used to build and test Java applications.

3. Eclipse IDE

<https://www.eclipse.org/>

Another popular IDE for Java development with support for Swing-based applications.

Other Resources:

1. Java Tutorials on YouTube (Java Programming Tutorials)

Various channels provide video tutorials that help users understand basic Java concepts, such as **The Net Ninja**, **Programming with Mosh**, and **Derek Banas**.