# A black and red text Description automatically generated

**PROJECT AND TEAM INFORMATION**

## Project Title

|  |
| --- |
| **Visualizing City-to-City Shortest Routes: A GUI-Based Implementation of Dijkstra’s Algorithm**. |

## Student/Team Information

|  |  |
| --- | --- |
| Team Name:  Team # (Mentor needs to assign) |  |
| Team member 1 (Team Lead)  Singh Devansh  230211141  [singhdevansh5001@gamil.com](http://singhdevansh5001@gmail.com) |  |
| Team member 2  (Last Name, name: student ID: email, picture): |  |
| **Team member 3:**  Gauri Guliyani  230221455  guliyanigauri@gmail.com |  |

**PROJECT PROGRESS DESCRIPTION (35 pts)**

## Project Abstract (2 pts)

|  |
| --- |
| The Shortest Path Finder GUI is a Java Swing application designed to visualize the shortest path between cities in India. The application uses Dijkstra's algorithm to compute the shortest path, displaying animated step-by-step traversal on a graphical map. Users can select start and end cities from a dropdown menu, and the application shows the shortest path along with the total distance. The path animation highlights the traversed route in real time, enhancing user engagement and understanding. |

## Updated Project Approach and Architecture (2 pts)

|  |
| --- |
| i)The project is developed using Java Swing for the graphical user interface.  ii)The graph is represented as an adjacency list stored in a HashMap, where keys are city names, and values are lists of edges with destination and weight.  iii)The city coordinates are predefined to accurately represent locations on a 2D plane.  iv)Dijkstra's algorithm is implemented to calculate the shortest path.  v)Path visualization is handled by a custom GraphPanel class, where animations are managed using a Swing Timer to display the shortest path step by step.  vi)The graph is bidirectional, supporting both directions for each connection between cities. |

## 

## Tasks Completed (7 pts)

|  |  |
| --- | --- |
| Task Completed | Team Member |
| i)Graph data structure setup (adjacency list)  ii) Dijkstra's algorithm integration  iii) Updating and optimizing city coordinate mapping  iv) Path animation using Swing Timer  v) GUI Implementation using Java Swing  vi) Adding obstacles dynamically using ActionListener  vii) Implementing error handling for invalid inputs (e.g., same start and end city) | Team Member 1  Team Member 1  Team Member 3  Team Member 2  Team Member 2  Team Member 3  Team Member 1 |

## Challenges/Roadblocks (7 pts)

|  |
| --- |
| i)GUI Responsiveness: Integrating path animation without freezing the UI required using the Swing Timer instead of Thread.sleep.  ii)Path Accuracy: Ensuring the correct visualization of the shortest path on a map required accurate city coordinates.  iii)Data Structure Complexity: Managing the graph as a bidirectional structure while maintaining efficient pathfinding.  iv)Path Animation: Achieving smooth, real-time animation with consistent frame rates.  v)Code Optimization: Balancing computation speed and visual appeal during path visualization. |

## Tasks Pending (7 pts)

|  |  |
| --- | --- |
| Task Pending | Team Member (to complete the task) |
| 1. Adding more cities to the graph 2. Improving GUI responsiveness 3. Adding distance labels on the map | Team Member 1  Team Member 2  Team Member 3 |

## 

## Project Outcome/Deliverables (2 pts)

|  |
| --- |
| i)A fully functional Shortest Path Finder GUI that visually displays the shortest path between selected cities using Dijkstra's algorithm.  ii)Interactive visualization with animated path highlighting.  iii)A user-friendly interface allowing city selection and displaying results. |

# Progress Overview (2 pts)

|  |
| --- |
| 1. The core functionalities (pathfinding and visualization) are completed. 2. The GUI is functional, but some enhancements are still pending, such as adding more cities and improving user experience. 3. Behind Schedule: GUI improvements and additional city data. 4. Ahead of Schedule: Algorithm integration and path animation. |

# Codebase Information (2 pts)

|  |
| --- |
| Repository link: <https://github.com/Devanshsingh31/Shortest_Path_Finder> Branch: main |

## Testing and Validation Status (2 pts)

|  |  |  |
| --- | --- | --- |
| Test Type | Status (Pass/Fail) | Notes |
| 1. Pathfinding Accuracy 2. GUI Responsiveness 3. Animation Flow | Pass  Pass  Pass | Verified with multiple city pairs  No lag during path animation  Path displays corectly in real time. |

# Deliverables Progress (2 pts)

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | **Deliverable**   1. Core GUI Functionality 2. Path Animation 3. Enhanced City Database 4. User Experience Enhancements | **Status**  Completed  Completed  In Progress  Pending | |