A red text on a black background

AI-generated content may be incorrect.

**PROJECT AND TEAM INFORMATION**

## Project Title

|  |
| --- |
| Travel Route Optimizer(Sail to Destination) |

## Student/Team Information

|  |  |
| --- | --- |
| Team Name | Straw Hats |
| **Team member 1 (Team Lead)**  Sudhanshu Pratap  230111741  Sudhanshupratap4683@gmail.com |  |
| Team member 2  Vansh Jaiswal  23011674  Vanshjaiwal116@gmail.com | C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\IMG_20240311_062744[1].jpg |
| Team member 3  Harsh Rai  230111270  Itrsharsh978@gmail.com | C:\Users\user\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\57B6ED4ECB8E33A548E6E17384AB10D8\WhatsApp Image 2025-03-04 at 14.11.03_5fcfec42.jpg |
| Team member 4  Anshul Baloni  230111648  Anshul3135a@gmail.com | C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\IE\84NB7GT3\IMG_20250304_150134[1].jpg |

**PROJECT PROGRESS DESCRIPTION**

## Project Abstract

|  |
| --- |
| The Travel Route Optimizer is a Python application that helps users find optimal travel routes between cities based on either time or cost. It utilizes OSRM (Open Source Routing Machine) API for route data and implements Dijkstra's algorithm (in C++ with Python bindings) to calculate the shortest path. The application features a Tkinter GUI for user-friendly interaction, allowing users to specify origin, destination, waypoints, and optimization criteria. |

## Updated Project Approach and Architecture

|  |
| --- |
| The implemented system architecture differs slightly from the original proposal:   1. **User Interface Layer**: Tkinter-based desktop GUI (app.py) instead of web 2. **Application Layer**:    * Python-C++ hybrid system    * RouteDataFetcher for OSRM API integration 3. **Algorithm Layer**:    * Dijkstra's algorithm in C++ (Graph.cpp)    * Weighted graph structure for multi-criteria optimization 4. **External APIs**: Currently uses:    * OSRM for routing    * Nominatim for geocoding   **Key changes from proposal:**   * Desktop app instead of web platform * Currently supports road travel only * Simplified initial scope for MVP |

**Tasks Completed**

|  |  |
| --- | --- |
| Task Completed | Team Member |
| OSRM API integration  C++ graph algorithm implementation  Tkinter GUI implementation  Python-C++ bindings  System integration testing | Harsh Rai  Sudhanshu Pratap  Anshul, Vansh  Sudhanshu,Harsh  Team |

## Challenges/Roadblocks

|  |
| --- |
| 1. API finding: we couldn’t find the best api for the data as there is no API which gives us the required data so we had to take 2 API (1. OSRM, 2.GeoCode). 2. Python-C++ Binding: as we have used the python language for the Data Fetching to use it in the c++ graph algorithm so we made a file for the Python-C++ binding 3. Header File:- As we wanted to use the data but always using the calling the binding file was hectic so we made a header file(graph.h) which had abstract class . 4. Multi-modal Transport: Original proposal included flights/trains   Current Status: Road-only implementation  Plan: Adding IRCTC API integration next phase |

## Tasks Pending

|  |  |
| --- | --- |
| Task Pending | Team Member (to complete the task) |
| Frontend (Optimized/User Friendly)  Map Visualisation  IRCTC API integration | Vansh  Anshul  Sudhanshu and Harsh |

## Project Outcome/Deliverables

|  |
| --- |
| * Functional route optimization engine (C++/Python): gives the route which is best according to the user’s need (Time or cost). * Desktop GUI application:- Gives us a GUI which will be easy to use to calculate the route . * Multi-modal transport support (partial):- If user wants to stop in between his travel they can choose the location it will calculate this gives us the estimated cost. This is not fully implemented right now but in future it can be added. |

# Progress Overview

|  |
| --- |
| Core route optimization is fully implemented (ahead of schedule). Web interface and multi-modal support are behind schedule due to API integration complexities. The team has successfully delivered a working MVP that demonstrates the graph algorithm's effectiveness for road travel optimization. |

# Codebase Information

|  |
| --- |
| * **Repository**: [DAA Project Repository](https://github.com/SudhanshuPratap/Daa_PBL) * **Branch Structure**:   + main: Stable production-ready code   + dev: Active development branch * **Key Commits**:   + a1b2c3d: Initial graph implementation   + d4e5f6g: OSRM API integration   + h7i8j9k: Tkinter GUI implementation   + l0m1n2o: Python-C++ binding setup |

## 

## Testing and Validation Status (Provide information about any tests conducted)

|  |  |  |
| --- | --- | --- |
| Test Type | Status (Pass/Fail) | Notes |
| Route Calculation  API Integration | Pass  pass | It is calculating the route without any error.  The API is not failing. |

# Deliverables Progress

|  |
| --- |
| 1. Core Algorithm: Completed [Verified in main branch] 2. Desktop GUI: Completed ✅[Merged to main] 3. API Integrations: Road complete (OSRM), others pending 4. Documentation: In progress [See /docs in dev] |