A red text on a black background

AI-generated content may be incorrect.

**PROJECT AND TEAM INFORMATION**

## **Project Title**

|  |
| --- |
| **Optifly** : Optimizing Airline Route for passenger’s efficiency and cost Saving. |

## **Student/Team Information**

|  |  |
| --- | --- |
| Team Name: **Tech Resolutions**  Team Mentor: **Miss. Swati Joshi** |  |
| Team member 1 (Team Lead)  **Abhay Kanojia**  student ID: 230115137  email: [abhaykanojia51@gmail.com](mailto:abhaykanojia51@gmail.com) |  |
| Team member 2  **Anvesha**  student ID: 230122228  email: [dipirawat830@gmail.com](mailto:dipirawat830@gmail.com) |  |
| Team member 3  **Tanmay Chauhan**  student ID: 23011566  email: [chauhantanmay253@gmail.com](mailto:chauhantanmay253@gmail.com) |  |

**PROJECT PROGRESS DESCRIPTION (35 pts)**

## **Project Abstract (2 pts)**

(Brief restatement of your project’s main goal. Max 300 words).

|  |
| --- |
| OptiFly is an intelligent airline route optimization system designed to compute the most efficient routes between cities based on various factors like cost, travel time, and layovers. The system addresses inefficiencies in current route-finding methods by creating a graph-based backend coupled with efficient shortest-path algorithms. It offers personalized route suggestions via a user-friendly interface and enables administrators to manage flight data dynamically. **Planned enhancements:** a real-time weather forecast module to inform users about weather conditions on their travel date. |

## **Updated Project Approach and Architecture (2 pts)** (Describe your current approach, including system design, communication protocols, libraries used, etc. Max 300 words).

|  |
| --- |
| Our project uses a **modular architecture**. The backend is developed in Java, managing data using local databases and representing flights and airports as a directed weighted graph. Pathfinding is handled using Dijkstra’s and Bellman-Ford algorithms.  **Java libraries used:** java.sql, java.util, java.time.  Two frontend pages developed using HTML, CSS, and JavaScript:   * **Client page:** Source, destination, and priority (time, cost, or both) input * **Admin page:** Add, update, and suspend flights   **Next steps:** connecting frontend to Java backend via REST APIs or middleware. Future plan includes integrating a real-time weather forecast feature. |

## 

## **Tasks Completed (7 pts)** (Describe the main tasks that have been assigned and already completed. Max 250 words).

|  |  |
| --- | --- |
| Task Completed | Team Member |
| 1. Server-side backend (airport data, flight management terminal, Conversion of airports into codes for ease in construction of graph) 2. Client-side backend (graph setup, pathfinding) 3. Frontend pages (client and admin interfaces) | * Abhay Kanojia * Tanmay Chauhan * Anvesha Rawat |

## **Challenges/Roadblocks (7 pts)** (Describe the challenges that you have faced or are facing so far and how you plan to solve them. Max 300 words).

|  |
| --- |
| 1. **Language Switch & Integration:** Integrating Java backend with web frontend posed difficulties due to language differences and lack of native support for direct interaction. 2. **Airport Data Compilation:** Sourcing and assigning reliable, structured codes to Indian airports required significant manual work. 3. **Library Limitations:** Java’s limited web integration libraries made linking frontend and backend more complex. 4. **Planned Resolutions:** Use Spring Boot or Java REST services for bridging, verified dataset preprocessing, and third-party APIs for weather forecasts. 5. **Lack of Real-time data:** Due to safety reasons no company is providing the future data of a flight which was the main component of the project. |

## **Tasks Pending (7 pts)** (Describe the main tasks that you still need to complete. Max 250 words).

|  |  |
| --- | --- |
| Task Pending | Team Member (to complete the task) |
| 1. Connect frontend to Java backend 2. Integrate real-time weather forecast 3. Final testing and optimization | * Tanmay Chauhan, Anvesha Rawat * Abhay Kanojia * Entire Team |

## 

## **Project Outcome/Deliverables (2 pts)**

(Describe what are the key outcomes / deliverables of the project. Max 200 words).

|  |
| --- |
| 1. **Route optimization software identifying optimal flights based on multiple criteria:** The project aims to deliver a comprehensive **route optimization software** capable of identifying the most efficient flight routes between two cities based on multiple user-defined criteria such as travel cost, flight duration, and number of layovers. This will empower travelers to make better-informed decisions suited to their preferences. 2. **Graph-based backend with real-time flight management:** A graph-based backend system has been implemented, where airports are represented as nodes and flights as edges within a directed weighted graph. This structure allows for efficient management of flight connections and enables future integration of shortest path algorithms for route optimization. 3. **User-friendly client and admin web interfaces:** The system also features **user-friendly web interfaces** for both clients and administrators. The client interface allows travelers to input their preferences and view flight details, while the admin interface provides tools to add, update, or suspend flights within the system database in real time. 4. **Real-time weather forecast display feature (planned):** a planned enhancement is the integration of a **real-time weather forecast feature**. This will allow users to check expected weather conditions on the scheduled flight date, enabling them to make safer, more reliable travel plans and factor weather risks into their route selection. |

# Progress Overview (2 pts) (Summarize how much of the project is done, what's behind schedule, what's ahead of schedule. Max 200 words.)

|  |
| --- |
| * **COMPLETED TASK**  1. The **backend logic for flight management** has been successfully developed. This includes creating a directed graph data structure representing airports and their connecting flights, alongside a functional server-side terminal that allows the administrator to add, update, and suspend flight details dynamically within the system’s local database. 2. On the **frontend side**, two fully functional web pages have been created — one for clients to enter travel preferences such as source, destination, and priority criteria (time, cost, or both), and another for the administrator to manage flight records in real time.  * **PENDING TASK**  1. The **remaining major tasks** include integrating a suitable **route optimization algorithm** (like Dijkstra’s or Bellman-Ford) into the backend for computing the most efficient flight routes. Additionally, establishing a seamless **connection between the frontend and Java backend** via REST APIs or middleware is pending, which will enable real-time interaction between the user interface and backend data operations. 2. An upcoming enhancement is the inclusion of a **real-time weather forecast feature**, providing weather updates for the selected flight date. While the integration of the optimization algorithm and frontend-backend connection is slightly behind schedule, all other project modules are progressing as planned. |

# Codebase Information (2 pts) (Repository link, branch, and information about important commits.)

|  |
| --- |
| Maintained locally. Repository link will be shared after integration. **Key commits:** Graph data structure creation, admin and user terminal setup, and frontend page development. |

## **Testing and Validation Status (2 pts)** (Provide information about any tests conducted)

|  |  |  |
| --- | --- | --- |
| Test Type | Status (Pass/Fail) | Notes |
| Backend graph creation and management | Pass | Flight data added and modified successfully |
| Admin terminal operations | Pass | Adds, updates, and suspends flights correctly |
| Frontend input forms | Pass | Captures client and admin inputs as intended |
| Route optimization algorithm | Pending | |  | | --- | |  |  |  | | --- | | To be implemented | |
| Frontend-backend connection | Pending | Planned in the next development phase |

# Deliverables Progress (2 pts) (Summarize the current status of all key project deliverables mentioned earlier. Indicate whether each deliverable is completed, in progress, or pending.)

|  |
| --- |
| 1. Graph-based flight management module Completed 2. Client terminal interaction Completed 3. Admin terminal interaction Completed 4. Client and Admin web pages Completed 5. Route optimization algorithm integration Pending 6. Frontend-backend integration Pending 7. Real-time weather forecast page (new add-on) Pending |