PROJECT REPORT

ON

"AirTracker"

Bachelor of Computer Application

(Batch: 2021-2025)

Submitted To:

Mr. Mandeep Singh

(Computer Engineering and Application)

Submitted By:

Akshat Maheshwari(2115000111)

Pankaj Ajmera(2115000693)

Rishit Gupta(2115000850)

Sudhanshu Tripathi(2115001009)

GLA UNIVERSITY, MATHURA

(Affiliated to NAAC A+ Grade)

CERTIFICATE

This is to certify that we Pankaj Ajmera, Rishit Gupta, Sudhanshu Tripathi and Akshat of BTech(CSE) 5th Semester from GLA University, Mathura has presented this Mini project work entitled “Air Tracker”, an app in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology under our supervision and guidance.

ACKNOWLEDGEMENT

It is our proud privilege to express our profound gratitude to the entire management of GLA University and the teachers of the institute for providing us with the opportunity to avail ourselves of the excellent facilities and infrastructure. The knowledge and values inculcated have proved to be of immense help at the very start of my career. Special thanks to the Hon’ble Founder, GLA University, Mathura for having provided us with an excellent infrastructure.

I am grateful to Mr. Mandeep Singh for their astute guidance, constant encouragement and sincere support for this project work.

Sincere thanks to all my family members, seniors and friends for their support and assistance throughout the project.

TABLE OF CONTENTS

CHAPTER – 1

INTRODUCTION

1.1. Identifying the Need

In today's dynamic travel landscape, finding the best flight options based on prices is a crucial need. We justified this issue through extensive statistics and documentation, highlighting the challenges users face in making informed decisions about their flights. Our justification draws from real-world scenarios, presenting a clear consultancy problem that demands resolution.

To substantiate this need further, we conducted surveys and gathered user feedback, affirming that the identified problem resonates with actual travelers. Additionally, our investigation aligned with relevant contemporary reports from reputable agencies, underscoring the urgency and significance of addressing this issue.

1.2. Defining the Problem

The broad problem at hand is optimizing the flight search experience. This involves providing users with a platform, AirTracker, that enables them to easily search for flights by specifying their origin, destination, and travel dates, ultimately displaying a list of flights with their corresponding prices. Importantly, at this stage, we refrain from hinting at specific solutions, focusing solely on articulating the problem that needs resolution.

1.3. Identifying Tasks

To tackle this problem effectively, we identified and differentiated tasks required at various stages of the project. This includes the identification phase, building phase, and testing phase. These tasks serve as the foundation for structuring the report, delineating chapters, headings, and subheadings that guide our exploration and analysis.

1.4. Setting the Timeline

A well-defined timeline, depicted through a Gantt chart, outlines the project's milestones and deadlines. This visual representation helps us manage tasks efficiently, ensuring a systematic and timely execution of the project.

1.5. Organization of the Report

This report is organized to provide a comprehensive understanding of the AirTracker project. Each chapter contributes to the project's progression, starting with an exploration of the identified need and problem, followed by the detailed tasks involved in addressing the issue. The timeline sets the pace for the report, and readers can expect insights, analyses, and findings distributed across the chapters.

As we embark on this journey, the following chapters will unfold, offering a holistic view of the development and implementation of AirTracker.

CHAPTER - 2

LITERATURE REVIEW/BACKGROUND STUDY

2.1. Timeline of the Reported Problem

To understand the global context of flight search challenges, we delved into the timeline of the reported problem. Our investigation spanned across incidents worldwide, backed by documentary proof. This historical exploration helped us grasp the evolution of issues related to finding optimal flight options based on prices.

2.2. Proposed Solutions

We explored previous attempts to solve the challenges faced in the domain of flight search. This involved examining various proposed solutions, understanding their key features, effectiveness, and drawbacks. By reviewing existing solutions, we aimed to learn from past experiences and identify opportunities for improvement in the development of AirTracker.

2.3. Bibliometric Analysis

A comprehensive bibliometric analysis was conducted to evaluate existing literature. This analysis focused on key features, effectiveness, and drawbacks of solutions implemented globally. By synthesizing this information, we gained insights into the current state of flight search applications, informing our approach to building an innovative solution with AirTracker.

2.4. Review Summary

Linking our findings from the literature review, we summarized key insights that directly relate to the development of AirTracker. This connection between existing knowledge and our project's objectives ensured that we were building on valuable lessons learned from the broader field of flight search applications.

2.5. Problem Definition

With a clear understanding of the historical context and existing solutions, we defined the problem AirTracker aimed to address. This involved specifying what needed to be done, how we intended to do it, and establishing boundaries for the project. The problem definition became a guiding framework for the subsequent design and development stages.

2.6. Goals/Objectives

Aligned with the problem definition, we set specific and measurable goals for AirTracker. These goals served as milestones during the project, providing a roadmap for our work. The objectives were designed to be narrow, precise, tangible, and concrete, ensuring that each step contributes directly to the success of AirTracker and can be validated or measured.

By conducting a thorough literature review and aligning our project objectives with the insights gained, AirTracker is positioned to offer an innovative and effective solution in the realm of flight search applications.

CHAPTER – 3

DESIGN OF THE SYSTEM

3.1 Planning Flight Search with React

To create an engaging and responsive user interface, we leveraged React, HTML, Tailwind-CSS, and JavaScript for the frontend. Users can easily search for flights by specifying their origin, destination, and travel dates. The application, designed with React components, dynamically displays a list of flights with corresponding prices.

3.2 Setting Up Authentication with Node.js and Firebase

For user authentication, we utilized Node.js for backend development and Firebase as the authentication and database system. Users can register and log in securely using their email and password. Node.js ensures a robust backend, while Firebase handles authentication seamlessly, making it a reliable choice for user management.

3.3 Integrating APIs for Dynamic Data

To provide real-time flight information, we integrated APIs into the backend. This enables AirTracker to fetch and display up-to-date flight details. Node.js acts as the middleware, handling requests and responses between the frontend and external APIs, ensuring a dynamic and accurate flight search experience.

3.4 Collaborative Development with Git and GitHub

For smooth collaboration and version control, we employed Git and GitHub. Multiple developers can work on the project simultaneously, tracking changes, and ensuring a stable codebase. This collaborative approach helps maintain consistency and facilitates efficient development.

3.5 Coding in Visual Studio Code

The actual coding process took place in Visual Studio Code, providing a streamlined and developer-friendly environment. This powerful code editor enhances productivity, offering features like syntax highlighting, debugging, and extensions, making the development process efficient and enjoyable.

3.6 Testing and Iterating

After coding, we thoroughly tested AirTracker to ensure its functionality and responsiveness. Testing involved simulating various user scenarios, validating authentication security, and confirming the accuracy of flight information. Any identified issues were addressed, and improvements were iteratively implemented.

3.7 Creating the Implementation Plan

Throughout the design process, we created a comprehensive implementation plan, mapping out the integration of frontend and backend technologies, API connections, and the authentication system. This plan guided the development, ensuring a cohesive and well-executed project.

By combining these technologies and features, AirTracker is designed to offer users a seamless and secure experience, allowing them to easily search for flights and manage their accounts.

CHAPTER - 4

RESULTS ANALYSIS AND VALIDATION

4.1 Implementation of Solution

When building AirTracker, we used some helpful tools to make sure the app works smoothly and provides the best flight options based on prices. Here's a simple breakdown of what we did:

4.2 Understanding the Data

We used modern tools to look at the information we get from the flight API. This helped us figure out things like how flight prices change and what users like. It's like looking at a big picture to make AirTracker better.

4.3 Drawing the Plan

Before building AirTracker, we drew pictures and plans to see how it should look and work. These drawings helped us understand how users will use the app and make sure it's easy for them to find and book flights.

4.4 Writing Reports

We wrote reports to explain everything we did. These reports talk about why we made certain choices, what problems we faced, and how we fixed them. These reports help us remember what we did and can guide us in the future.

4.5 Keeping Things Organized

We used tools to organize our work and talk to each other. This made sure everyone knew what to do and helped us solve problems faster. Good communication is like the glue that keeps everything together.

4.6 Checking and Testing

Before showing AirTracker to you, we tested it a lot. We pretended to be users, tried different things, and made sure everything worked as expected. This testing helped us be confident that AirTracker gives you the best flight options.

The work we did in this phase made AirTracker possible. We used the API to get data, and these steps helped us create a simple and effective React app that shows you the best flight options based on prices.

CHAPTER - 5

CONCLUSION & FUTURE WORK

5.1 Conclusion

In conclusion, the AirTracker project represents a forward-thinking web application designed to empower users in their quest for the most affordable flights and real-time ticket price tracking. Leveraging a robust technological stack including React, HTML, Tailwind-CSS, JavaScript, Node.js, Firebase, and APIs, the project is poised to redefine the flight search experience.

Anticipated to be completed within a month, the AirTracker team is driven by the vision of delivering a user-friendly and efficient web application. By enabling users to seamlessly search for the cheapest flights and track ticket prices, the project aligns with the evolving needs of modern travelers.

The team is confident in achieving the project objectives and is committed to delivering a fully functional AirTracker web application. This confidence stems from a comprehensive understanding of user requirements, cutting-edge technologies, and a collaborative development approach.

5.2 Future Work

As AirTracker enters the post-development phase, several avenues for future work come into focus:

5.2.1. Feature Enhancement

Continued refinement of AirTracker can involve feature enhancement. The team aims to iterate on the application, incorporating additional functionalities that enhance the overall user experience. This may include refining search algorithms, introducing personalized user preferences, and expanding the scope of tracked flight information.

5.2.2. Technological Advancements

To stay at the forefront of innovation, future work will entail staying abreast of technological advancements. This involves periodic updates to the technological stack, ensuring that AIrTracker remains compatible with the latest frameworks and industry standards. The integration of emerging technologies may also be explored for further optimization.

5.2.3. User Community Engagement

A critical aspect of future work is engaging with the user community. Establishing a feedback loop and actively seeking user insights will be prioritized. This iterative approach allows for continuous improvements based on user feedback, ensuring that AirTracker evolves in tandem with user expectations.

5.2.4. Global Expansion

Looking beyond the initial release, there is potential for global expansion. Future efforts may focus on integrating additional airlines, supporting a broader range of destinations, and adapting the application to cater to diverse international markets.

In summary, the future work for AirTracker encompasses a commitment to ongoing improvement, technological currency, user engagement, and potential global scalability. By embracing these aspects, AirTracker aims to not only meet but exceed user expectations in the dynamic landscape of flight search applications.

REFERENCES

1. React - A JavaScript library for building user interfaces. (https://reactjs.org/)
2. HTML – Hyper Text Markup Language, the standard markup language for documents designed to be displayed in a web browser. (https://html.spec.whatwg.org/multipage/)
3. Tailwind CSS - A utility-first CSS framework. (https://tailwindcss.com/)
4. JavaScript - A high-level, interpreted programming language that conforms to the ECMAScript specification. (https://developer.mozilla.org/en-US/docs/Web/JavaScript)
5. Node.js - An open-source, cross-platform JavaScript runtime environment for executing JavaScript code server-side. (https://nodejs.org/)
6. Firebase - A platform developed by Google for creating mobile and web applications. (https://firebase.google.com/)
7. APIs - Application Programming Interfaces, used for communication between different software systems.

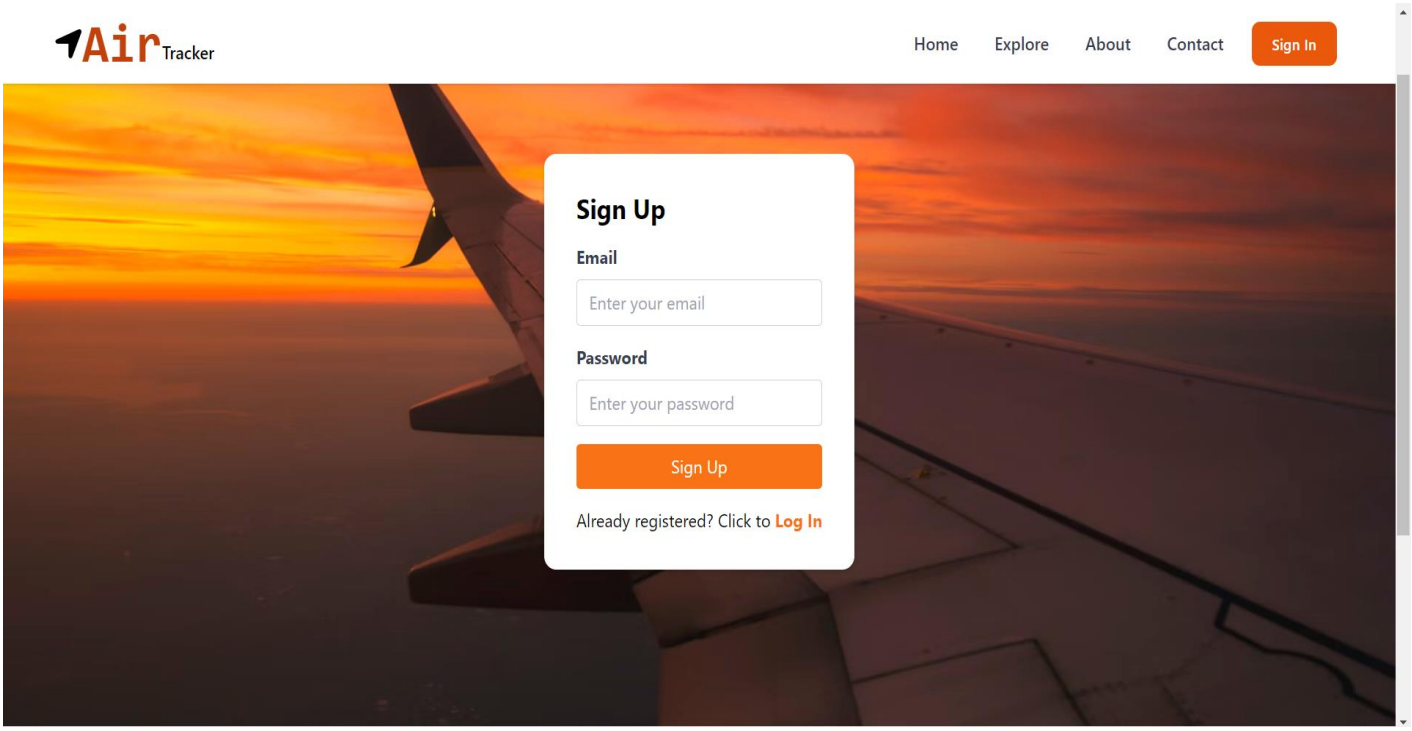
The AirTracker team acknowledges the invaluable contributions of the above technologies, frameworks, and resources that played a pivotal role in the conceptualization and development of the AirTracker web application.

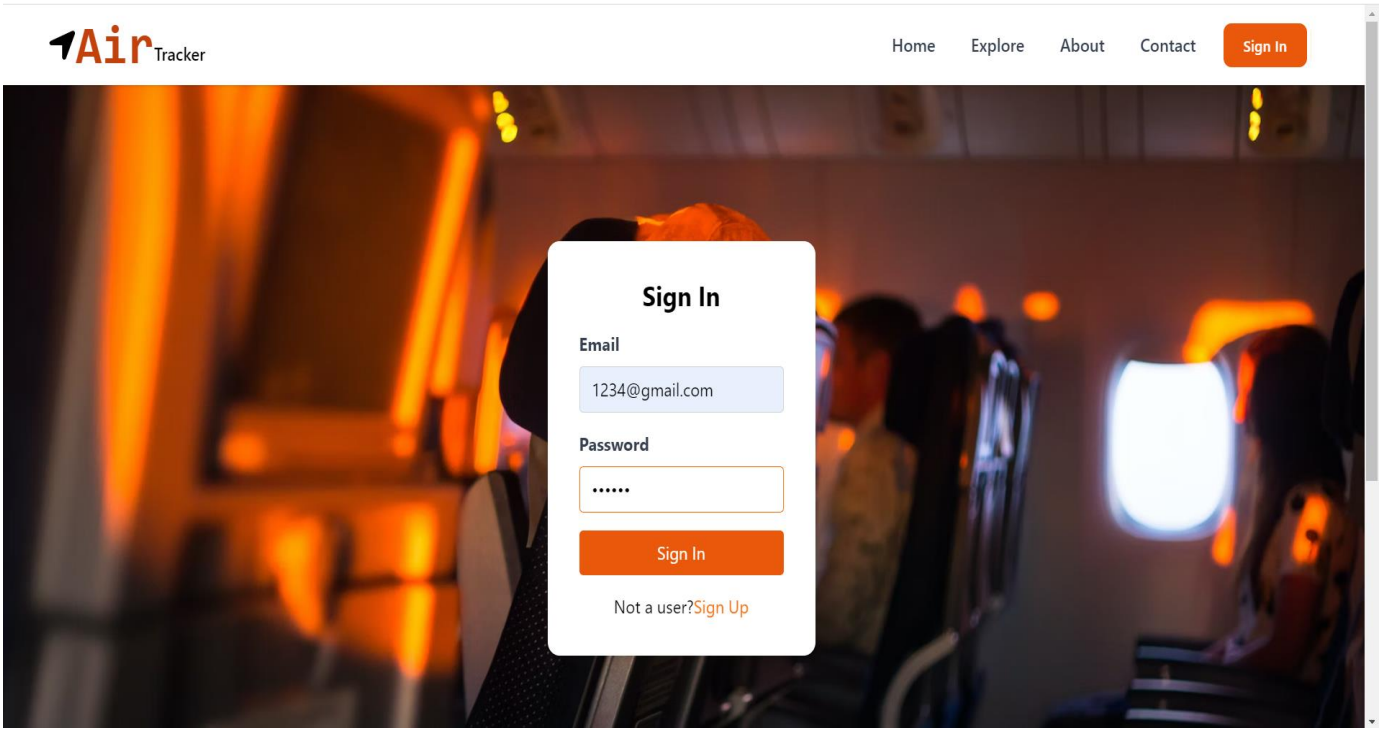
Github Link :- https://github.com/sudhanshu-77/AirTracker

APPENDIX

User Manual

```
PS C:\PROJECT\api\new> git clone https://github.com/sudhanshu-77/AirTracker
Cloning into 'AirTracker'...
remote: Enumerating objects: 140, done.
remote: Counting objects: 100% (140/140), done.
remote: Compressing objects: 100% (96/96), done.
Receiving objects: 100% (140/140), 2.87 MiB | 789.00 KiB/s, done.

PS C:\PROJECT\api\new\AirTracker> cd AirTracker01
PS C:\PROJECT\api\new\AirTracker\AirTracker01> npm i

added 466 packages, and audited 467 packages in 20s

111 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\PROJECT\api\new\AirTracker\AirTracker01> npm run dev

> airtracker01@0.0.0 dev
> vite


  VITE v5.0.0  ready in 359 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```


Air Tracker
Home
Explore
About
Contact
Sign In
Sign Up
Email
Enter your email
Password
Enter your password
Sign Up
Already registered? Click to Log In


Air Tracker
Home
Explore
About
Contact
Sign In
Sign In
Email
1234@gmail.com
Password
Sign In
Not a user?Sign Up

Air Tracker

Home Explore About Contact LogOut

Fly More, Pay Less

Flight Search

Source

Destination

Departure

dd-mm-yyyy

Class

Economy

Traveler

1 0 0

Air Tracker

Home Explore About Contact Sign In

Best Results

Departure Time 08:20
Departure Date 2023-11-30
Departure Airport Code DEL
Airlines Code UKAI
Class: ECONOMY

15h 15m
Duration

Arrival Time 23:35
Arrival Date2023-11-30
Arrival Airport Code BOM
₹735
Book

Departure Time 05:45
Departure Date 2023-11-30
Departure Airport Code DEL
Airlines Code 6E
Class: ECONOMY

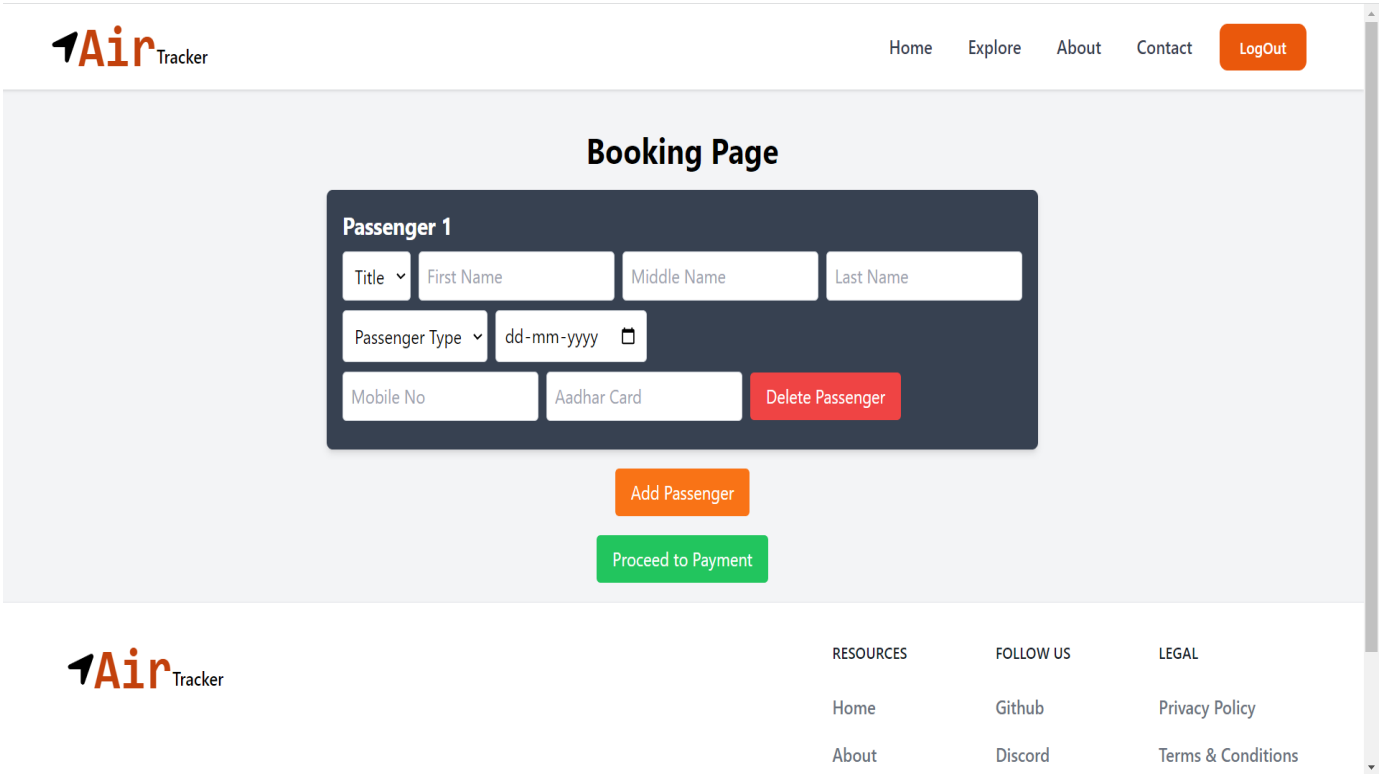
04h 35m
Duration

Arrival Time 10:20
Arrival Date2023-11-30
Arrival Airport Code BOM
₹868.5
Book

Departure Time 19:35
Departure Date 2023-11-30
Departure Airport Code DEL
Airlines Code I5
Class: ECONOMY

08h 55m
Duration

Arrival Time 04:30
Arrival Date2023-12-01
Arrival Airport Code BOM
₹880.33
Book



Air Tracker

RESOURCES	FOLLOW US	LEGAL
Home	Github	Privacy Policy
About	Discord	Terms & Conditions