

Industry Project Report

On

Travel rate Scrapping

Developed by:

Nirva_Patel (20162101014)

Guided By:

Prof. Ravindra Patel (Internal)

Mr. Pratik Patel (External)

Submitted to

**Department of Computer Science & Engineering Institute of
Computer Technology**



**Ganpat
University**

॥ विद्यया समाजोत्कर्षः ॥

**Institute of
Computer
Technology**



Year: 2024



**Ganpat
University**
॥ विद्यया समाजोत्कर्षः ॥

**Institute of
Computer
Technology**



CERTIFICATE

This is to certify that the **Industry Project** work entitled “**Travel Rate Scraping**” by Patel Nirva Nileshkumar (20162101014) of Ganpat University, towards the fulfilment of requirements of the degree of Bachelor of Technology – Computer Science and Engineering, carried out in the CSE(CBA) Department at OneClick IT Consultancy Pvt. Ltd. The results/findings contained in this Project have not been submitted in part or full to any other University / Institute for award of any other Degree/Diploma.

Name & Signature of Internal Guide

Name & Signature of Head

Place: ICT - GUNI

Date:

ACKNOWLEDGEMENT

IBM project is a golden opportunity for learning and self-development. I consider myself very lucky and honored to have so many wonderful people lead me through in completion of this project. First and foremost, I would like to thank Dr. Rohit Patel, Principal, ICT, and Prof. Dharmesh Darji, Head, ICT who gave us an opportunity to undertake this project. My grateful thanks to Prof. Ravi Patel & Mr. Harsh Bantuya (Internal & External Guides) for their guidance in project work Online Blockchain based certificate generation and validation system for government organization, who despite being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where would have been without his/her help. CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Patel Nirva Nileshekumar (20162101014)

ABSTRACT

The travel rate scraping project proposes an automated system to streamline flight data collection from various B2B and B2C websites into a centralized database. It aims to offer users comprehensive flight information, including the cheapest fares and all available options for specific routes, accessible through an intuitive dashboard. Future work includes expanding data sources, integrating advanced data analysis techniques, and incorporating user preferences for personalized recommendations. Real-time updates and mobile application development are suggested to enhance accessibility, while partnerships with travel agencies and airlines can improve service quality and access to exclusive deals. User feedback mechanisms will drive iterative improvements, ensuring the system remains responsive to evolving user needs and industry trends. Through these enhancements, the system can evolve into a robust platform, providing travelers with efficient and informed flight booking solutions.

INDEX

CHAPTER 1: INTRODUCTION.....	1
1.1 Introduction.....	2
CHAPTER 2: OBJECTIVE.....	3
2.1 Primary Objective	4
2.2 Literature review	4
CHAPTER 3: METHODOLOGY.....	5
3.1 Website selection	6
3.2 Technology selection	6
3.3 Web scraping.....	6
3.4 Database management	7
3.5 Frontend Development.....	7
CHAPTER 4: IMPLEMENTATION DETAILS.....	8
4.1 Web scraping.....	9
4.2 Database management	10
4.3 JSON Data handling	10
4.4 Error handling and Scalability	11
CHAPTER 5: SYSTEM DESIGN.....	12
5.1 Data flow diagram.....	13
5.2 Use Case diagram	14
5.3 ER diagram	15

CHAPTER 6: REQUIREMENTS.....	16
6.1 Hardware and Software requirements.....	17
6.1.1 Hardware requirements.....	17
6.1.2 Software requirements	17
6.2 Additional tools and Libraries.....	18
 CHAPTER 7: SCREENSHOT OF DEVELOPMENT PROCESS	19
7.1 Screenshots of ixigo.....	20
7.2 Screenshots of Makemytrip	23
7.3 Screenshots of Scrapped and Stored data	26
 CHAPTER 8: CONCLUSION AND FUTURE WORK.....	32
8.1 Conclusion	33
8.2 Future work	33
 CHAPTER 9: REFERENCES.....	34

CHAPTER 1: INTRODUCTION

CHAPTER 1: INTRODUCTION

1.1 Introduction

- ✓ The flight data scraping project introduces an innovative solution for efficiently gathering flight information from various online platforms, both business-to-business (B2B) and business-to-consumer (B2C). The goal is to centralize this data into a single database, providing users with a comprehensive view of available flights, including pricing details and route options. The project's primary objective is to simplify the process of finding the best flight deals by presenting all relevant information in an easy-to-use dashboard. By automating data collection and aggregation, users can quickly compare fares and make informed decisions.
- ✓ Future developments for the project include expanding the sources of data to ensure coverage of a wider range of airlines and travel agencies. Additionally, plans involve incorporating advanced data analysis techniques to offer users personalized recommendations based on their preferences and travel history. To enhance accessibility, the project proposes the development of a mobile application, providing real-time updates and allowing users to access flight information on the go. Furthermore, partnerships with travel agencies and airlines are envisaged to improve service quality and provide users with access to exclusive deals and promotions.
- ✓ Continuous feedback mechanisms will be implemented to gather user input and refine the system based on evolving needs and industry trends. This iterative approach ensures that the system remains responsive and adaptable to changing requirements.
- ✓ Overall, the project aims to evolve into a robust platform that offers travelers efficient and informed solutions for booking flights. By leveraging technology and user feedback, the system will continue to enhance the travel booking experience, making it easier and more convenient for users to find the best flight options available.

CHAPTER 2: OBJECTIVE

CHAPTER 2: OBJECTIVE

2.1 Primary Objective

The primary objective of this project is to develop a travel rate scraping system that collects flight data from diverse sources and stores it into a database.

Specifically, the objectives include:

- Implementing web scraping algorithms to extract flight information from B2B and B2C websites.
- Designing a database schema to efficiently store and manage the scraped data.
- Developing a user-friendly dashboard interface to display the cheapest flights and all available options for a given route.
- Implementing algorithms to analyze and compare flight prices, enabling users to make informed decisions.
- Ensuring the scalability and reliability of the system to accommodate future expansions and updates.

2.2 Literature Review

Previous research in the field of travel technology has explored various approaches to automate the process of collecting and analyzing flight data. Web scraping techniques have been widely adopted to extract information from websites efficiently. Additionally, database management systems have been utilized to store and organize large volumes of data collected from disparate sources. Several studies have also focused on developing user interfaces that provide intuitive ways for travelers to search and compare flight options. By building upon existing methodologies and technologies, this project aims to contribute to the advancement of travel technology by providing a comprehensive and user-centric solution for accessing flight information.

CHAPTER 3: METHODOLOGY

CHAPTER 3: METHODOLOGY

3.1 Website selection

Flight data will be scraped from a diverse selection of B2B and B2C travel websites, considering factors like popularity, reliability, and relevance to users. Websites with large user bases, trusted data accuracy, and catering to various traveler preferences will be prioritized for data extraction.

3.2 Technology selection

- Choose appropriate technologies for web scraping, database management, and frontend development.
- Node.js with Crawlee (Playwright) is selected for web scraping due to its robustness and flexibility.
- MySQL is chosen for database management due to its reliability and scalability.
- Next.js is selected for building the frontend dashboard due to its efficiency in creating interactive and responsive web applications.

3.3 Web scraping

- ✓ Utilizing Crawlee (Playwright), a Node.js library, for web scraping offers several advantages. Playwright's comprehensive API allows for seamless automation of web interactions, enabling the scraping of flight information such as prices, departure times, airlines, and available routes from selected websites.
- ✓ Developing scraping scripts with error handling mechanisms is essential to ensure the reliability and robustness of the scraping process. Error handling mechanisms should address potential issues such as website changes, page loading errors, and data inconsistencies to ensure smooth operation and accurate data extraction.
- ✓ By implementing these strategies, the project can effectively scrape flight data from a variety of B2B and B2C travel websites, providing users with accurate and up-to-date information to facilitate informed decision-making.

3.4 Database management

- ✓ Creating a MySQL database schema to store scraped flight data involves designing tables to organize and manage the information effectively. Key tables may include those for flights, airlines, routes, and additional relevant data.
- ✓ For the flights table, attributes such as flight number, departure and arrival airports, departure and arrival times, and ticket prices can be included. Additionally, the airlines table can store information about different airlines, such as their names and contact details.
- ✓ Routes can be stored in a separate table, containing details about specific flight paths, including departure and arrival airports and distances. Other relevant information, such as aircraft types or departure terminals, can be stored in additional tables as needed.
- ✓ Utilizing the Node.js MySQL library, a connection with the database can be established to execute SQL queries for tasks such as data insertion, retrieval, and management. This allows for seamless integration between the backend and the database, enabling efficient handling of scraped flight data.
- ✓ Implementing data validation and normalization techniques is crucial to maintain data integrity and consistency within the database. This involves validating incoming data to ensure it meets predefined criteria and normalizing the data structure to eliminate redundancy and improve efficiency.

3.5 Frontend Development

- ✓ Develop the frontend dashboard using Next.js, a React framework for server-side rendering and client-side routing.
- ✓ Design intuitive user interfaces with responsive layouts and interactive components for displaying flight information.
- ✓ Implement features such as search functionality, sorting options, and filters to facilitate user navigation and exploration of flight data.

CHAPTER 4: IMPLEMENTATION DETAILS

CHAPTER 4: IMPLEMENTATION DETAILS

4.1 Web scraping

Node.js with Crawler (Playwright): Utilized Node.js with Crawler, a web scraping library built on top of Playwright, to automate the extraction of flight data from various B2B and B2C travel websites.

Playwright Crawler: Developed custom crawlers using Playwright to navigate through web pages, interact with elements, and extract flight information such as airline names, flight numbers, prices, and source-destination pairs.

Scraping Multiple Websites: Implemented scraping scripts for multiple websites to collect comprehensive flight data. Each script was tailored to the structure and layout of the respective website, ensuring accurate extraction of relevant information.

Integration of Playwright and Crawler: Leveraged Playwright's browser automation capabilities within Crawler's framework to orchestrate scraping tasks efficiently. Crawler abstracted away the complexities of Playwright, providing a higher-level interface for defining scraping logic and managing concurrency.

How Playwright and Crawler Work Together: Integrated Playwright's browser automation capabilities within Crawler's framework to automate browser interactions, navigation, and data extraction efficiently. Crawler abstracted away the complexities of Playwright, providing a higher-level interface for defining scraping tasks and managing concurrency.

4.2 Database management

MySQL for Data Storage: Employed MySQL as the relational database management system (RDBMS) for storing the scraped flight data. Designed a database schema comprising tables for flights, airlines, routes, and pricing information to organize the data efficiently.

Express App for API Endpoint: Implemented an Express.js application to serve as an API endpoint for interacting with the MySQL database. Defined RESTful routes and endpoints to handle CRUD operations for retrieving, updating, and deleting flight data.

Comparison of Prices: Developed algorithms to compare prices obtained from different airlines across various websites. Utilized unique airline numbers as identifiers to match and compare prices with corresponding entries in the database tables.

4.3 JSON Data Handling

Storage in JSON Files: Stored JSON data representing request bodies in files within the project directory. These JSON files contained structured data representing the parameters required for initiating scraping tasks.

Utilization in Crawlee: Loaded the JSON data from files and passed them as input parameters to Crawlee's scraping tasks. This facilitated the customization and configuration of scraping tasks based on specific requirements and user preferences.

4.4 Error handling and Scalability

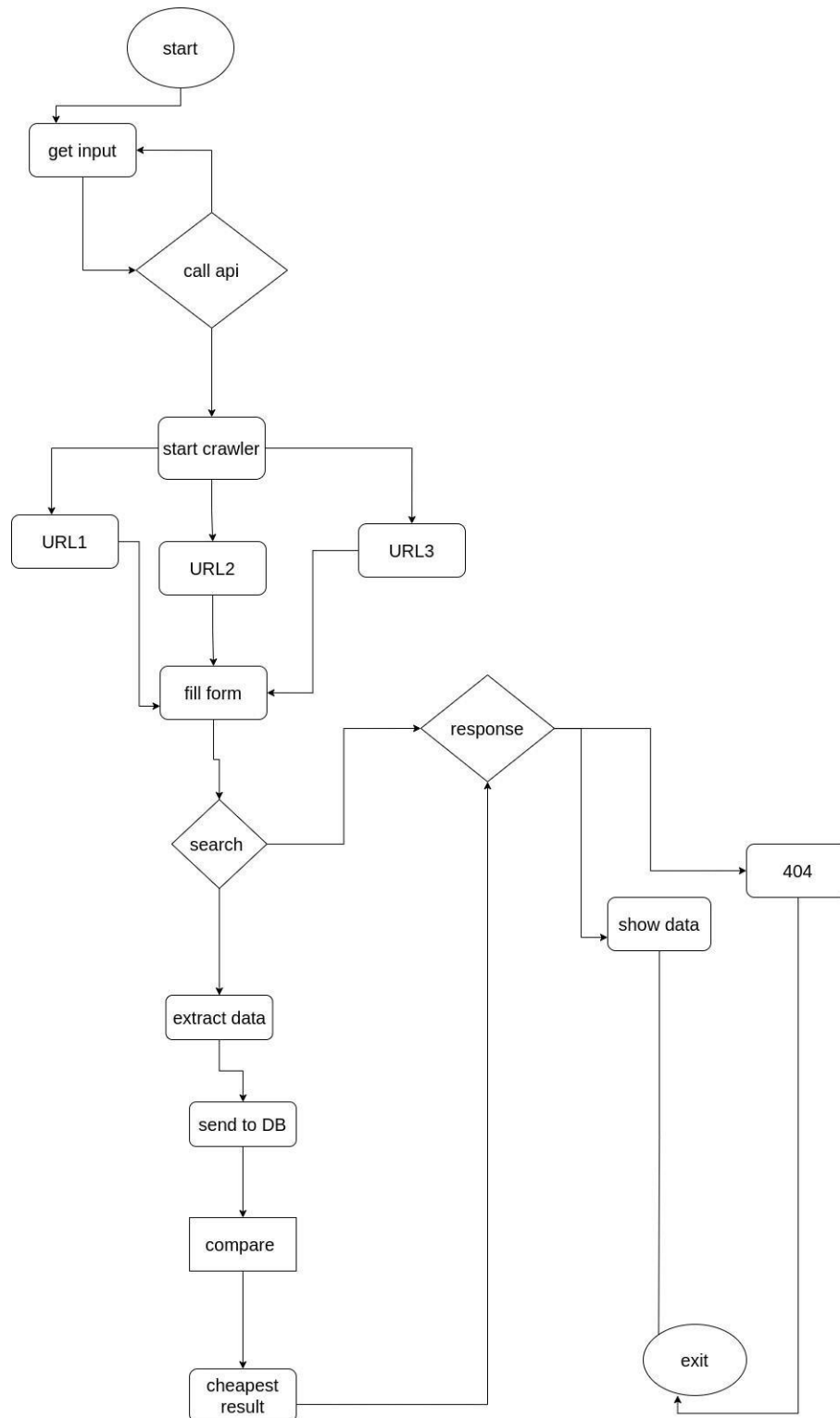
Error Handling: Implemented robust error handling mechanisms within the web scraping scripts to gracefully handle exceptions, such as network errors, page timeouts, and unexpected changes in website structure. Employed retry strategies and logging functionalities to capture and manage errors effectively, ensuring the reliability and resilience of the scraping process.

Scalability: Designed the architecture of the web scraping system to be scalable, capable of handling large volumes of data and concurrent scraping tasks. Utilized asynchronous programming techniques in Node.js to optimize performance and resource utilization, enabling the system to scale horizontally by adding more scraping instances or vertically by leveraging powerful hardware resources. Employed load balancing and distributed processing strategies to distribute scraping tasks across multiple servers or cloud instances, further enhancing the system's scalability and responsiveness to varying workloads.

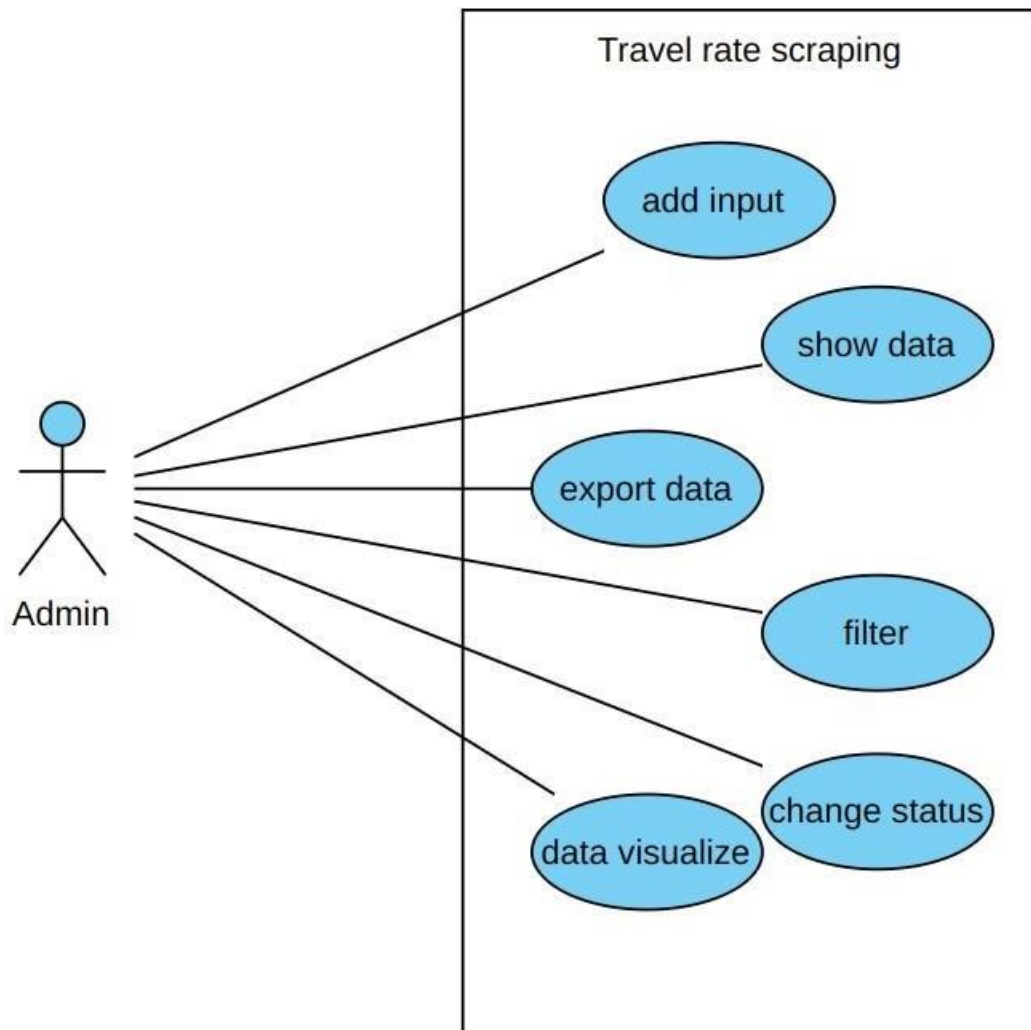
CHAPTER 5: SYSTEM DESIGN

CHAPTER 5: SYSTEM DESIGN

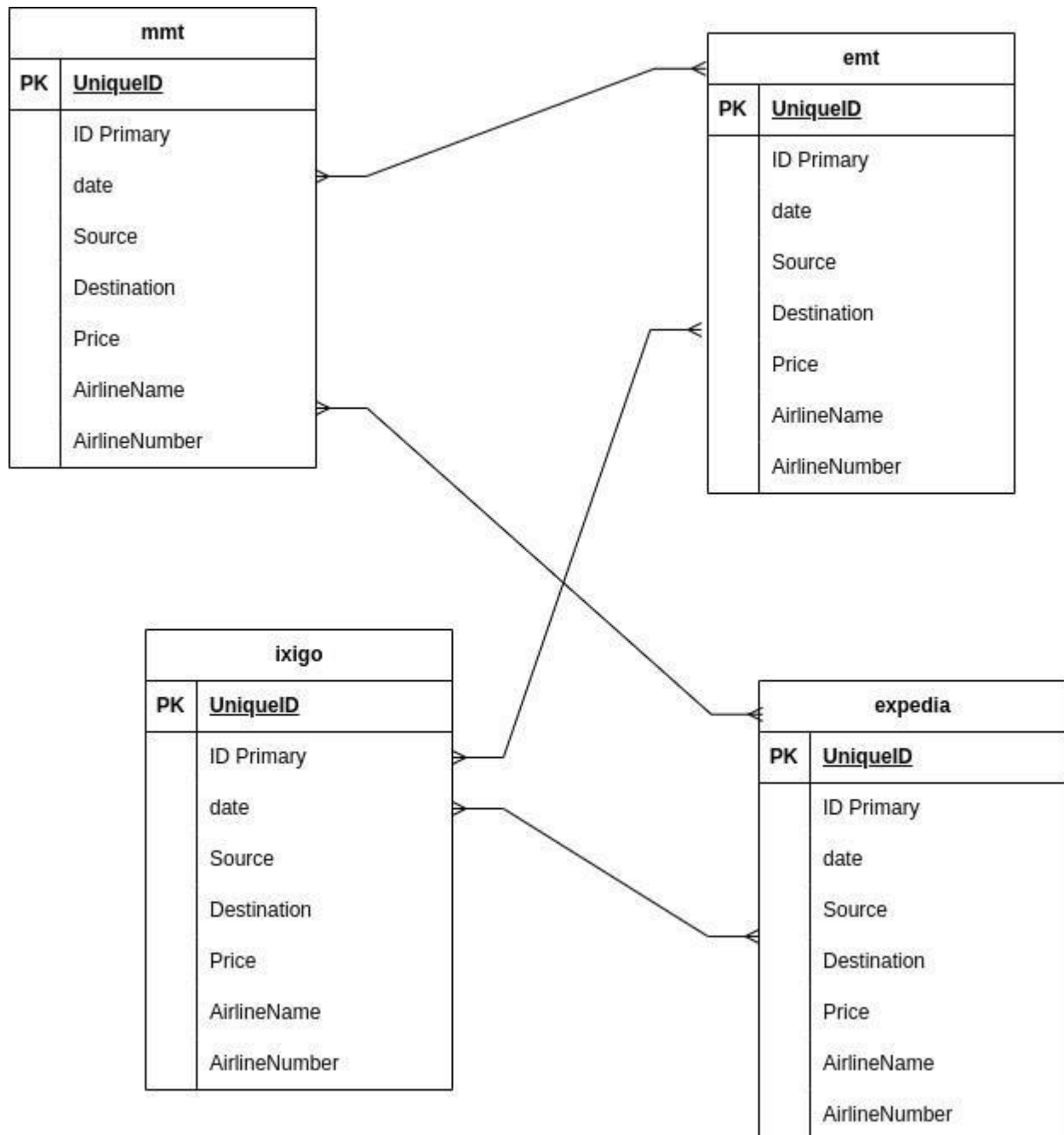
5.1 Data flow diagram



5.2 Use Case diagram



5.3 ER diagram



CHAPTER 6: REQUIREMENTS

CHAPTER 6: REQUIREMENTS

6.1 Hardware and Software requirements

6.1.1 Hardware requirements

Processor: Intel Core i5 or equivalent processor (or higher) for optimal performance.

RAM: Minimum 8GB RAM for smooth operation of development tools and database management.

Storage: At least 256GB SSD for storing project files, databases, and related data.

Network: Stable internet connection for web scraping tasks and accessing online resources.

6.1.2 Software requirements

Operating System: Any modern operating system supported by Node.js, such as Windows, macOS, or Linux.

Development Environment:

- Node.js: Latest stable version of Node.js installed for running JavaScript-based applications.
- Visual Studio Code or any preferred code editor for writing and editing project code.

Web Scraping Tools:

- Crawlee with Playwright: Install Crawlee library along with Playwright for web scraping tasks.

Database Management System:

- MySQL: Latest version of MySQL Community Server for storing and managing scraped flight data.

Frontend Development Framework:

- Next.js: Install Next.js framework for building the frontend dashboard interface.
- React.js: Next.js is built on top of React.js, so ensure React.js is also installed.

Backend Framework:

- Express.js: Install Express.js for creating API endpoints to interact with the MySQL database.

Dependency Management:

- npm (Node Package Manager): Used for installing and managing project dependencies.

Version Control:

- Git: Version control system for tracking changes to project code and collaborating with team members.
- GitHub or GitLab: Online platforms for hosting project repositories and facilitating collaborative development.

6.2 Additional tools and Libraries

Postman: Useful for testing API endpoints and making HTTP requests during development.

MySQL Workbench: GUI tool for managing MySQL databases and executing SQL queries.

Browser Developer Tools: Chrome DevTools or similar tools for inspecting web pages and debugging web scraping scripts.

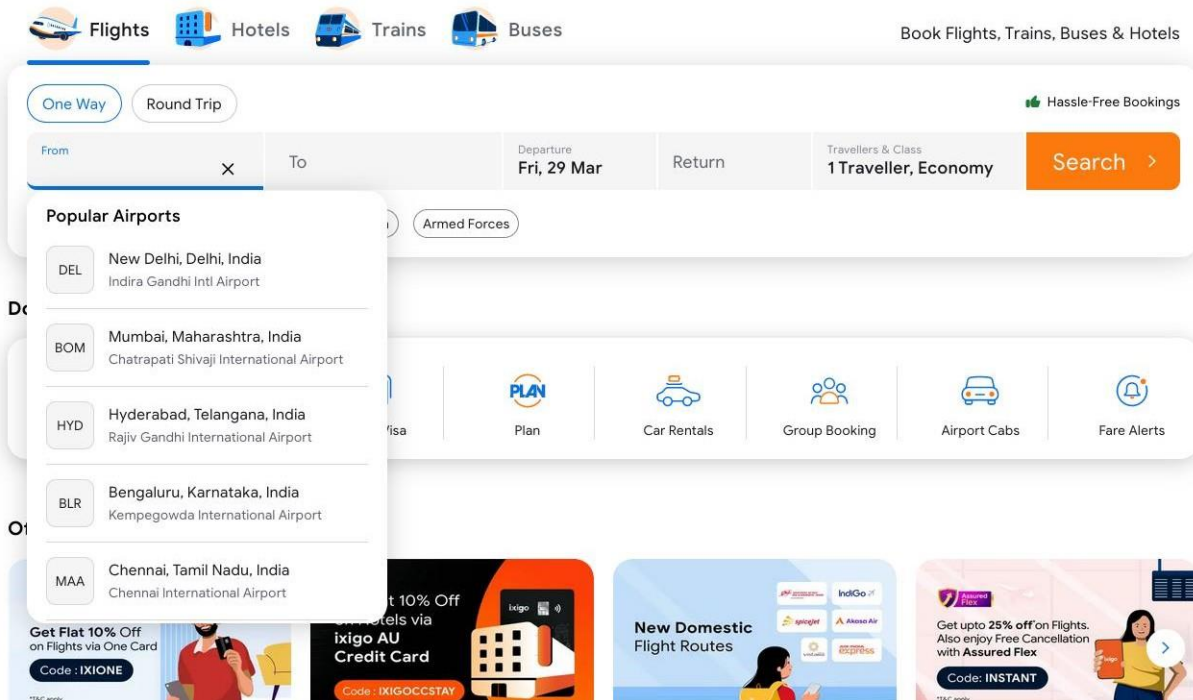
Data Visualization Libraries: Optional libraries such as D3.js or Chart.js for visualizing scraped flight data in the frontend dashboard.

CHAPTER 7: SCREENSHOT OF DEVELOPMENT PROCESS

CHAPTER 7: SCREENSHOT OF DEVELOPMENT PROCESS

7.1 Screenshots of ixigo

- Entering Departure Location using crawlee automation / playwright crawler.



- Entering Arrival Location using playwrite.

Flights Hotels Trains Buses Book Flights, Trains, Buses & Hotels

One Way Round Trip 24x7 Customer Support

From AMD - Ahmedabad To Dubai Departure Fri, 29 Mar Return Travellers & Class 1 Traveller, Economy Search >

Special Fares (Optional): Student DXB Dubai, Dubayy, United Arab Emirates Dubai International Airport

Do More With ixigo

Flight Status Credit Card Book Visa Plan Car Rentals Group Booking Airport Cabs Fare Alerts

Offers For You

onecard Get Flat 10% Off on Flights via One Card Code: IXIONE

Get Flat 10% Off on Hotels via ixigo AU Credit Card Code: IXIGOCSTAY

New Domestic Flight Routes

Get upto 25% off on Flights. Also enjoy Free Cancellation with Assured Flex Code: INSTANT

- Selecting departure Date on ixigo.

Flights Hotels Trains Buses Book Flights, Trains, Buses & Hotels

One Way Round Trip Hassle-Free Bookings

From AMD - Ahmedabad To DXB - Dubai Departure Fri, 29 Mar Return Travellers & Class 1 Traveller, Economy Search >

Special Fares (Optional): Student Senior Citizen Armed

Do More With ixigo

Flight Status Credit Card Book Visa

Offers For You

onecard Get Flat 10% Off on Flights via One Card Code: IXIONE

Get Flat 10% Off on Hotels via ixigo AU Credit Card Code: IXIGOCSTAY

May Day 01 May • 1 Day

Rabindranath Tagore Jayanti Long Weekend 04 May - 07 May • 4 Days

Holiday List >

- Extracting flight details from the searched flight results.

[Flights](#)
[Hotels](#)
[Trains](#)
[Buses](#)
[More](#)
[Offers](#)
[Customer Service](#)
[Log in/Sign up](#)

From
AMD - Ahmedabad

To
DXB - Dubai

Departure
Sun, 22 Dec

Return

Travellers & Class
1 Traveller, Economy

Search

Special Fares (Optional):

Student

Senior Citizen

Armed Forces

Filters

Stops

Non-Stop

1 Stop

2+ Stops

Exclude

Non-Refundable

Flight Price

21308

457626

Departure from Ahmedabad

Sat, 21 Dec

Sun, 22 Dec
₹21308

Mon, 23 Dec

Tue, 24 Dec
₹18965

Wed, 25 Dec
₹20775

Thu, 26 Dec
₹20776

Fri, 27 Dec
₹17061

Sort by

Price
Low to High

Fastest
Shortest First

Departure
Earliest First

Smart
Recommended

144 Flights Available

Cheapest

Air India
AI614, AI983

08:05
AMD

15h 20m
1 stop

21:55
DXB

~~₹21,308~~ ₹20,053

Book

Flight Details

Air India
AI638, AI983

20:30
AMD

26h 55m
1 stop

21:55⁺
DXB

~~₹21,308~~ ₹20,053

Book

Flight Details

Air India
AI638, AI909

20:30
AMD

15h 15m
1 stop

10:15⁺
DXB

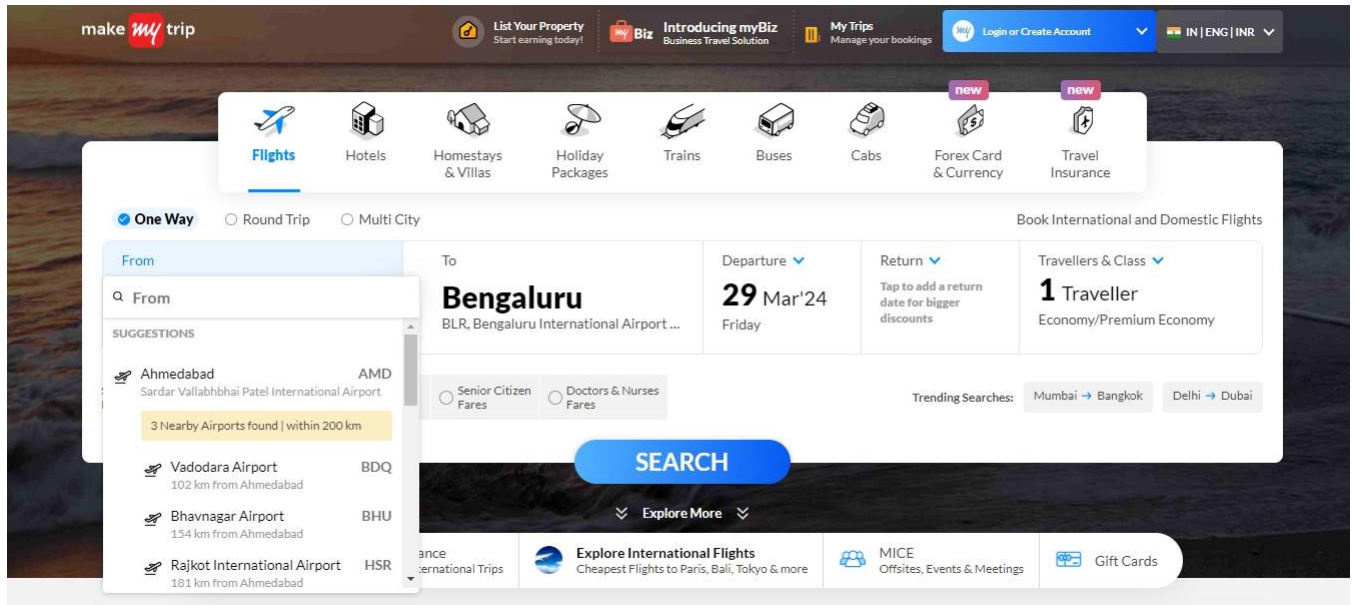
~~₹21,518~~ ₹20,258

Book

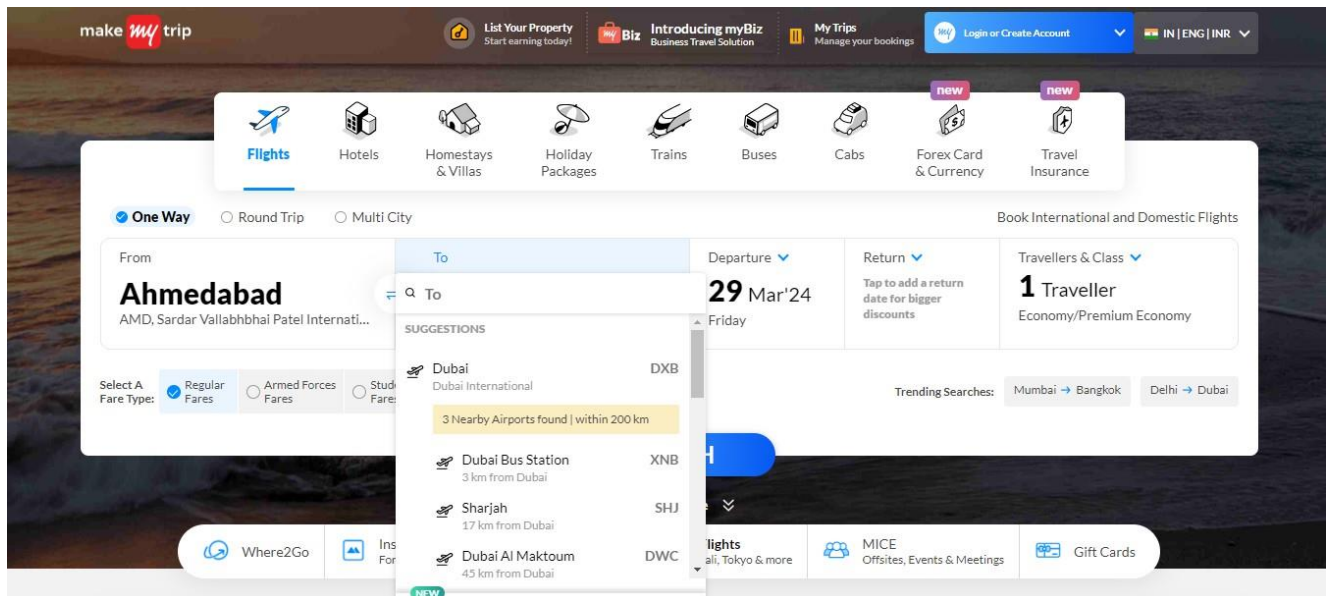
Flight Details

7.2 Screenshots of Makemytrip

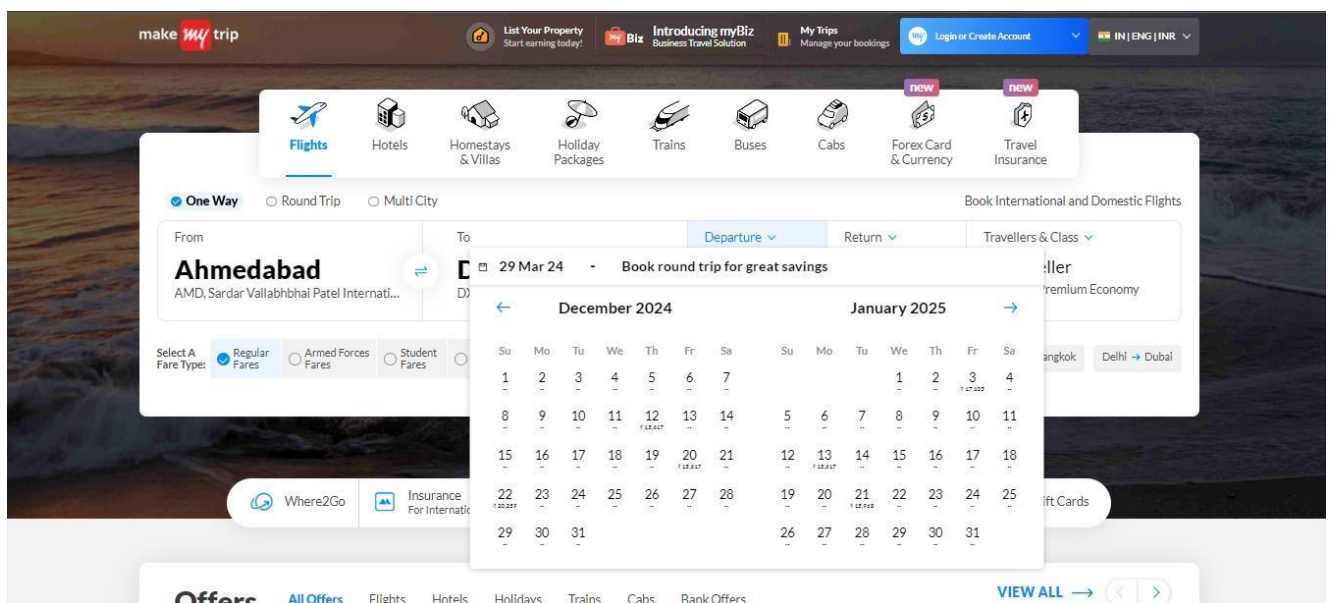
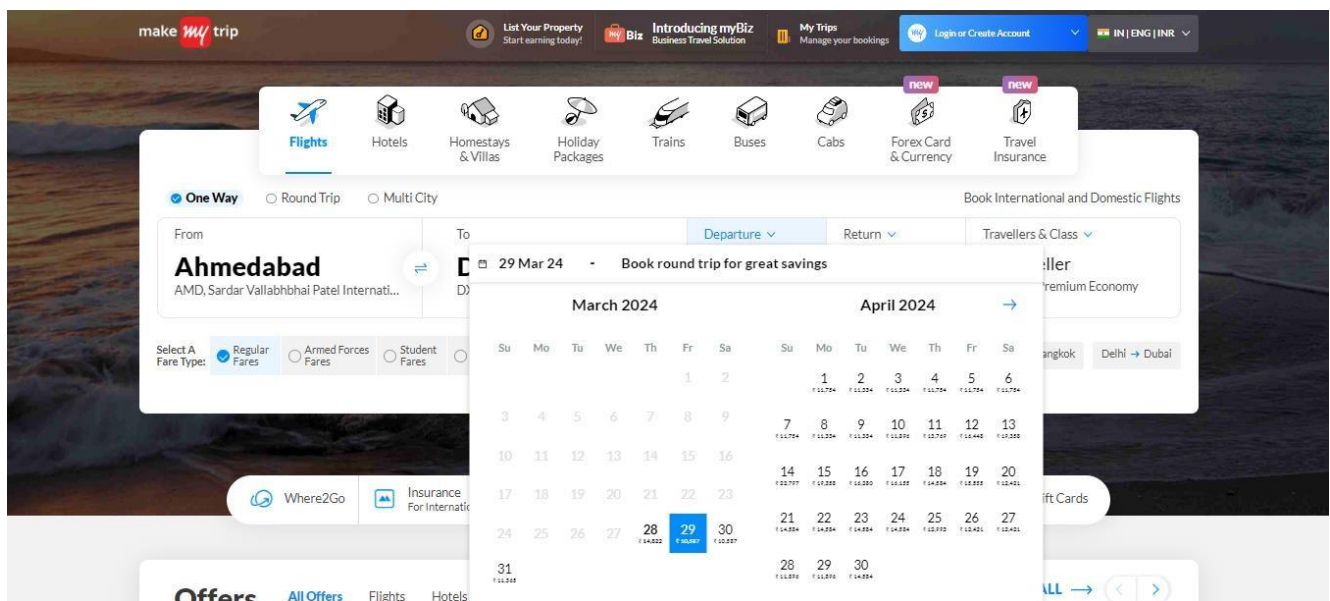
- Selecting the departure location.



- Selecting arrival location.



- Picking Date on mmt using playwright's page.click()



- Extracting searched flight details from MakeMyTrip.

make my trip

FlightsHotelsHomestays & VillasHoliday PackagesTrainsBusesCabsForex Card & CurrencyTravel InsuranceMore

Log in / Create Account

One Way

FROMAhmedabad, India

TODubai, United Arab E

DEPARTSun, Dec 22, 2024

RETURNSelect Return

PASSENGERS & CLASS1 Adult, Economy

SEARCH

Fare Type:RegularArmed ForcesStudentSenior CitizensDoctors & Nurses

Check-in Baggage Filter

Show flights with Check-in Baggage

Popular Filters

☐ Non Stop

₹ 20,724

☐ SpiceJet

₹ 26,895

☐ After Noon Departure

₹ 21,518

☐ Early Morning Departures

₹ 21,109

+ 5 more

Arrival Airports

☐ Abu Dhabi Intl (127Km)

₹ 20,724

☐ Sharjah (19Km)

₹ 21,109

☐ Ras Al Khaimah (77Km)

₹ 35,385

☐ Dubai International

₹ 21,308

One Way Price

₹ 20,724

₹ 1,21,000

Flights from Ahmedabad to Dubai

Thurs, Dec 19

Fri, Dec 20

Sat, Dec 21

Sun, Dec 22

Mon, Dec 23

Tue, Dec 24

Wed, Dec 25

Thu, Dec 26

CHEAPEST

₹ 20,724 | 02h 05m

NON STOP FIRST

₹ 20,724 | 02h 05m

YOU MAY PREFER

₹ 20,781 | 02h 15m

Flights sorted by Lowest fares on this route

Cheaper Non-stop Flights available on 20 Dec

Ethiad Airways

ET 553

10:15

Ahmedabad

02h 05m

Non stop

11:50

Abu Dhabi (127 Km from Dubai)

₹ 20,724

per adult

VIEW PRICES

Use code MMTDEALS to get Rs.699/person* off on this booking + Get additional Rs.150 off on UPI payment

Non Refundable

View Flight Details

Air Arabia AB...

3112

06:15

Ahmedabad

02h 15m

Non stop

08:00

Abu Dhabi (127 Km from Dubai)

₹ 20,781

per adult

VIEW PRICES

Use code MMTDEALS to get Rs.699/person* off on this booking + Get additional Rs.150 off on UPI payment

Non Refundable

View Flight Details

Looking for Cheaper Nonstop Flights?

- Qunar data scrap flight details.

Qunar B2B Distribution Portal

romitarora@itoneclick.com

sourceDEL Oneclick USD

Qunar B2B Distribution Portal User Manual.docx

Logout

Flight Search

MMB

User Manager

Depart CityAMD

Arrive CityDEL

Cabin LevelAll

Adult Number1

Child Number0

One-Way

Departure Date2024-05-29

Search

Round-Trip

AMD - DEL, 2024-05-29, results20

lowest price

earliest departure time

earliest arrival time

Shortest travel time

Stops

00

01

02+

All

Airlines

QF

6E

UK

SG

AI

CH

All

Go Trip

Depart Airport	Arrive Airport	Flight Number	Cabin Level	Cabin Code	Depart Time	Arrive Time	Stops	Transit	Price
AMD(T1)	DEL(T2)	QP1145	economy	K	2024-05-29 06:20	2024-05-29 07:55			From USD 56.63 Select
AMD(T1)	DEL(T2)	6E2385	economy	V	2024-05-29 23:20	2024-05-30 01:00			From USD 58.97 Select
AMD(T1)	DEL(T2)	QP1342	economy	K	2024-05-29 09:15	2024-05-29 10:55			From USD 62.15 Select
AMD(T1)	DEL(T3)	UK946	economy	V	2024-05-29 08:40	2024-05-29 10:25			From USD 63.95 Select
AMD(T1)	DEL(T2)	6E2736	economy	Q	2024-05-29 07:05	2024-05-29 08:40			From USD 63.53 Select
AMD(T1)	DEL(T3)	6E5079	economy	Q	2024-05-29 09:45	2024-05-29 11:15			From USD 67.26 Select
AMD(T1)	DEL(T2)	6E2032	economy	V	2024-05-29 18:55	2024-05-29 20:35			From USD 65.47 Select
AMD(T2)	DEL(T3)	SG8194	economy	U	2024-05-29 06:10	2024-05-29 07:50			From USD 55.52 Select

7.3 Screenshots of Scrapped and Stored data

- TBO flight data stored in DB.

The screenshot shows the MySQL Workbench interface. The left sidebar displays a schema tree with a 'scrap' database selected. The central query editor contains the SQL statement: `SELECT * FROM scrap.tbos;`. The result grid below the editor displays a table with columns: `id`, `searchedAt`, `departureLocation`, `arrivalLocation`, `departureDate`, `arrivalDate`, `flightNo`, `airline`, `price`, and `direct`. The table contains 20 rows of flight data. The right sidebar shows the 'SQL Additions' panel with 'My Snippets'.

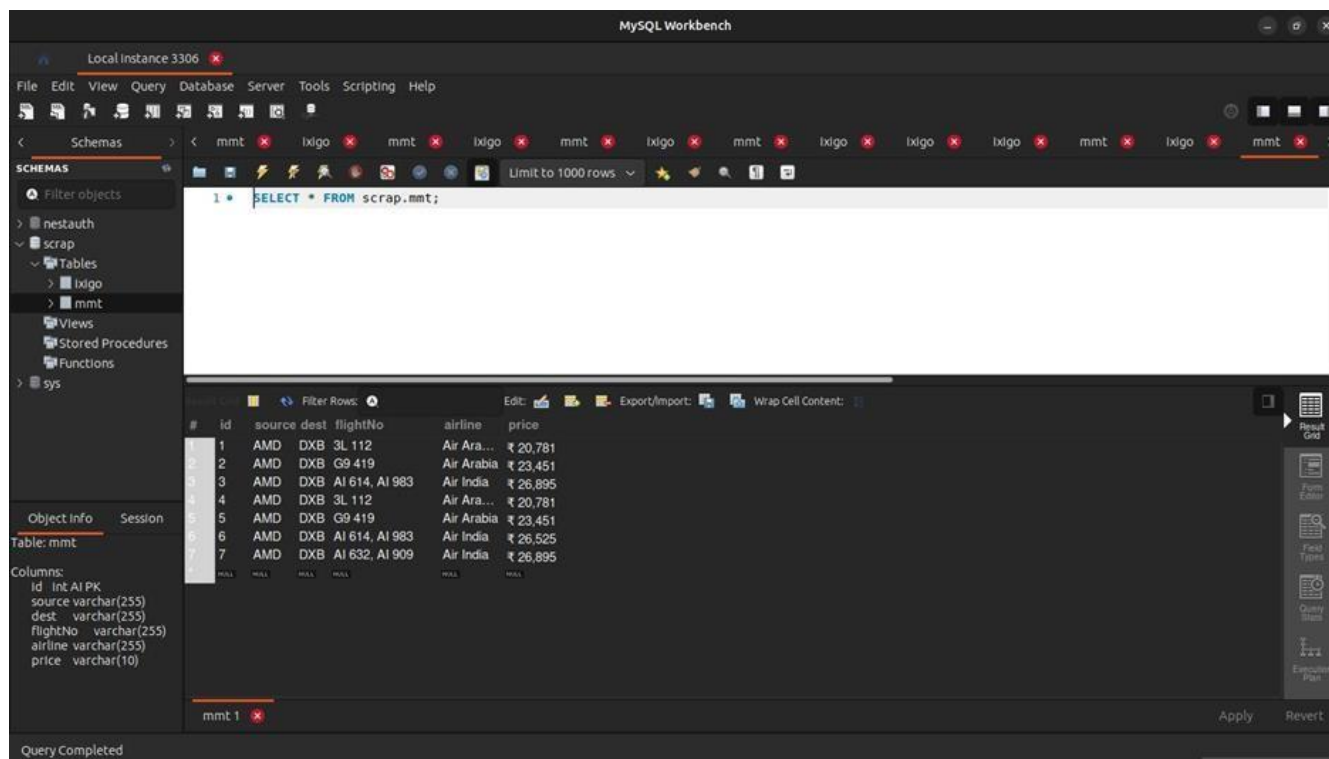
id	searchedAt	departureLocation	arrivalLocation	departureDate	arrivalDate	flightNo	airline	price	direct
1	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[2263]	[Indigo]	[[{"id": 1, "type": "Server", "price": 4720, "pub... 1	1
2	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[8194]	[SpiceJet]	[[{"id": 1, "type": "Coupon", "price": 5118, "pub... 1	1
3	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[2047]	[Indigo]	[[{"id": 1, "type": "Server", "price": 5252, "pub... 1	1
4	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[1145]	[Akasa Air]	[[{"id": 1, "type": "Server", "price": 5401, "pub... 1	1
5	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[976]	[Vistara]	[[{"id": 1, "type": "Cluster/TSP", "price": 5417, "pub... 1	1
6	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[2032]	[Indigo]	[[{"id": 1, "type": "Server", "price": 5510, "pub... 1	1
7	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[6042]	[Indigo]	[[{"id": 1, "type": "Server", "price": 5513, "pub... 1	1
8	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[482]	[Air India]	[[{"id": 1, "type": "Publish", "price": 5554, "pub... 1	1
9	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[1342]	[Akasa Air]	[[{"id": 1, "type": "Server", "price": 5590, "pub... 1	1
10	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[7568, 2029]	[Indigo]	[[{"id": 1, "type": "Server", "price": 5782, "pub... 0	0
11	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[5079]	[Indigo]	[[{"id": 1, "type": "Server", "price": 5826, "pub... 0	0
12	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[2736]	[Indigo]	[[{"id": 1, "type": "Server", "price": 5892, "pub... 0	0
13	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[906]	[Vistara]	[[{"id": 1, "type": "Cluster/TSP", "price": 5935, "pub... 0	0
14	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[726, 2162]	[Indigo]	[[{"id": 1, "type": "Server", "price": 6146, "pub... 0	0
15	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[7145, 2176]	[Indigo]	[[{"id": 1, "type": "Server", "price": 6146, "pub... 0	0
16	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[2794]	[Indigo]	[[{"id": 1, "type": "Server", "price": 6205, "pub... 0	0
17	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[946]	[Vistara]	[[{"id": 1, "type": "Cluster/TSP", "price": 6292, "pub... 0	0
18	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[836]	[Air India]	[[{"id": 1, "type": "Publish", "price": 6351, "pub... 0	0
19	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[6244, 6008]	[Indigo]	[[{"id": 1, "type": "Server", "price": 6661, "pub... 0	0
20	2024-04-23 21:14:26	AMD	DEL	2024-05-24	2024-05-24	[277, 546]	[Indigo]	[[{"id": 1, "type": "Server", "price": 6864, "pub... 0	0

- IXIGO flight scrapped data in MySQL

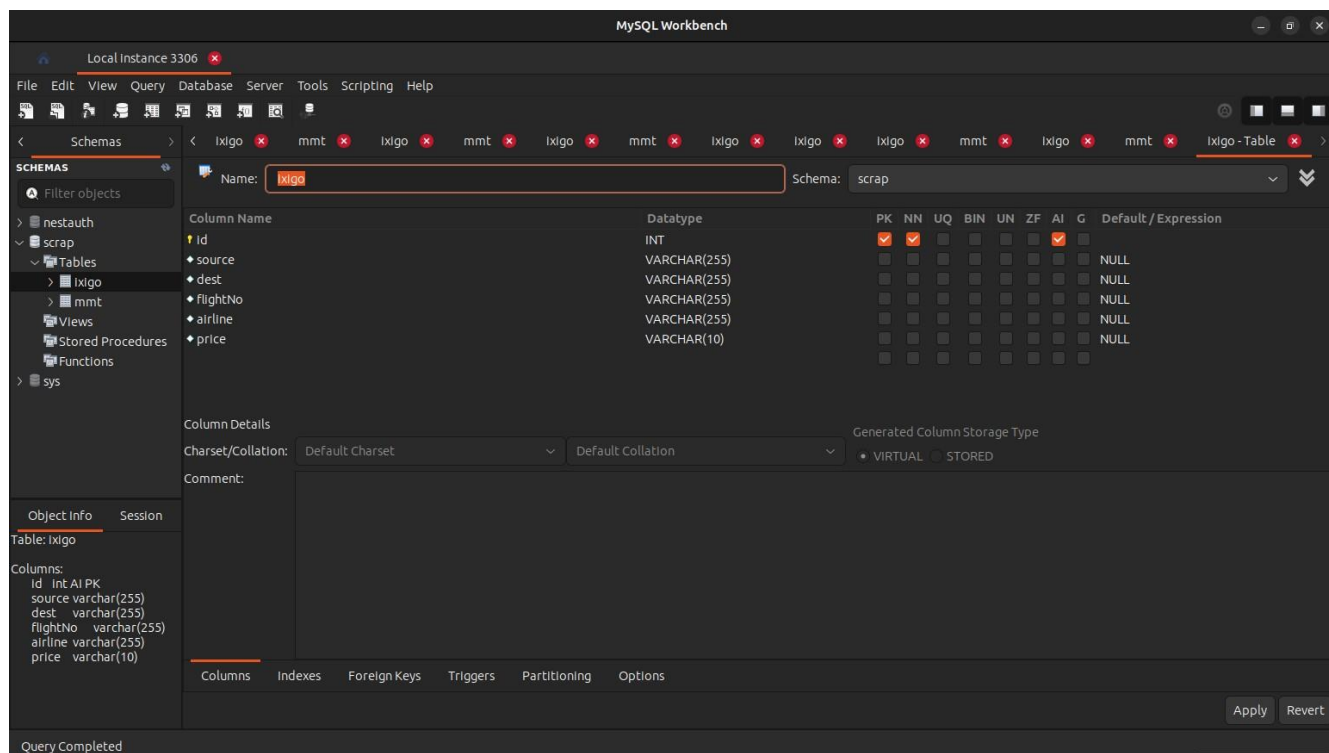
The screenshot shows the MySQL Workbench interface. The left sidebar displays a schema tree with a 'scrap' database selected. The central query editor contains the SQL statement: `SELECT * FROM scrap.ixigo;`. The result grid below the editor displays a table with columns: `id`, `source`, `dest`, `flightNo`, `airline`, and `price`. The table contains 14 rows of flight data. The right sidebar shows the 'SQL Additions' panel with 'My Snippets'.

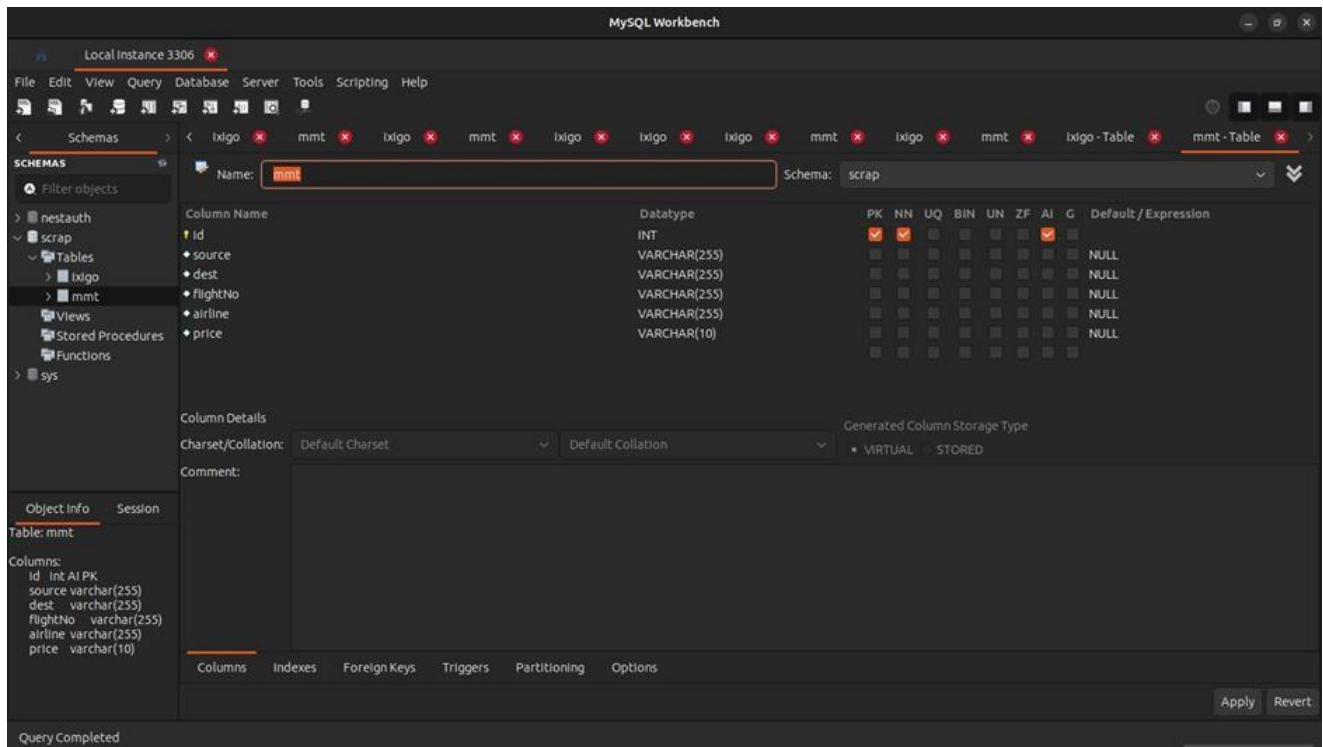
#	id	source	dest	flightNo	airline	price
1	AMD	DXB	AI638, AI983	Air India		₹20,053
2	AMD	DXB	AI638, AI909	Air India		₹20,258
3	AMD	DXB	AI632, AI909	Air India		₹20,258
4	AMD	DXB	AI614, AI909	Air India		₹20,258
5	AMD	DXB	EK541	Emirates		₹21,703
6	AMD	DXB	EK539	Emirates		₹21,703
7	AMD	DXB	UK918, AI983	Vistara, Air India		₹21,744
8	AMD	DXB	AI482, AI947	Air India		₹21,858
9	AMD	DXB	AI532, AI929	Air India		₹21,858
10	AMD	DXB	AI836, AI929	Air India		₹21,858
11	AMD	DXB	AI482, AI929	Air India		₹21,858
12	AMD	DXB	UK918, AI909	Vistara, Air India		₹21,949
13	AMD	DXB	AI482, AI995	Air India		₹21,958
14	AMD	DXB	AI532, AI995	Air India		₹21,958

- Storing MMT scraped data in MySQL.

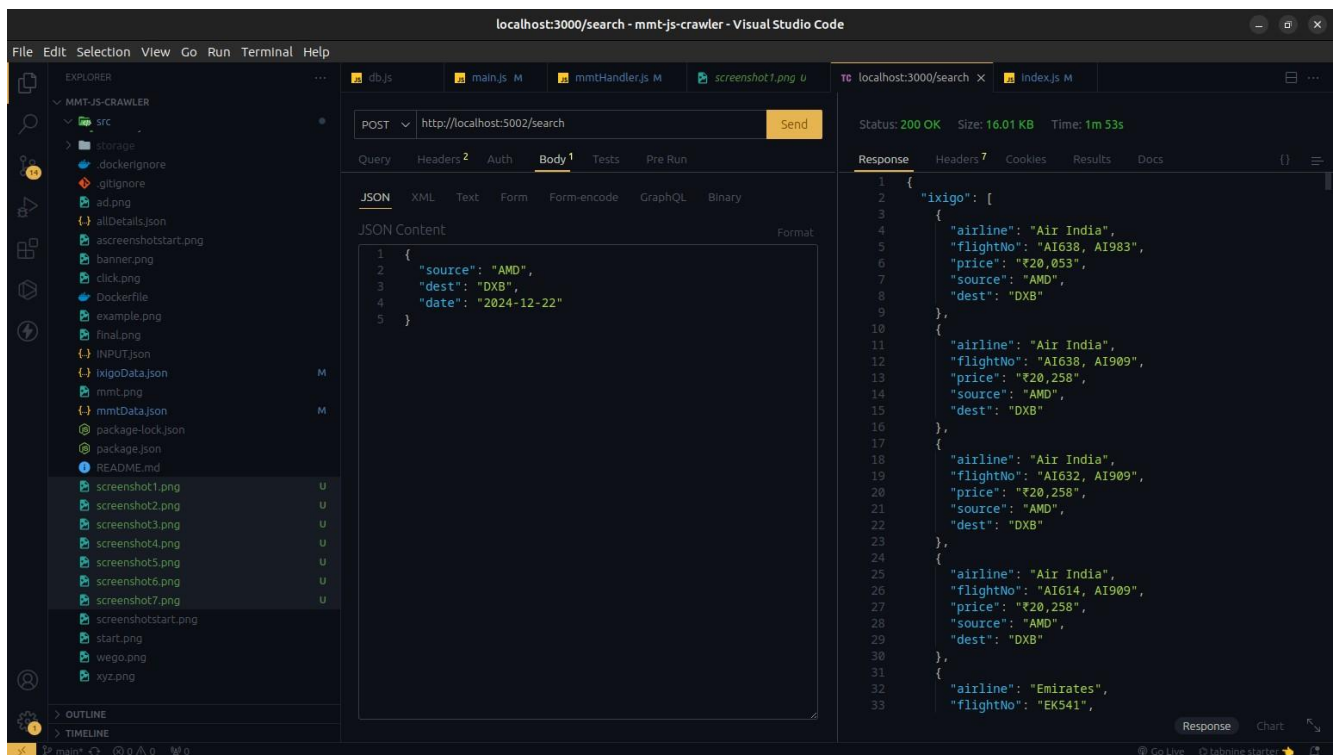


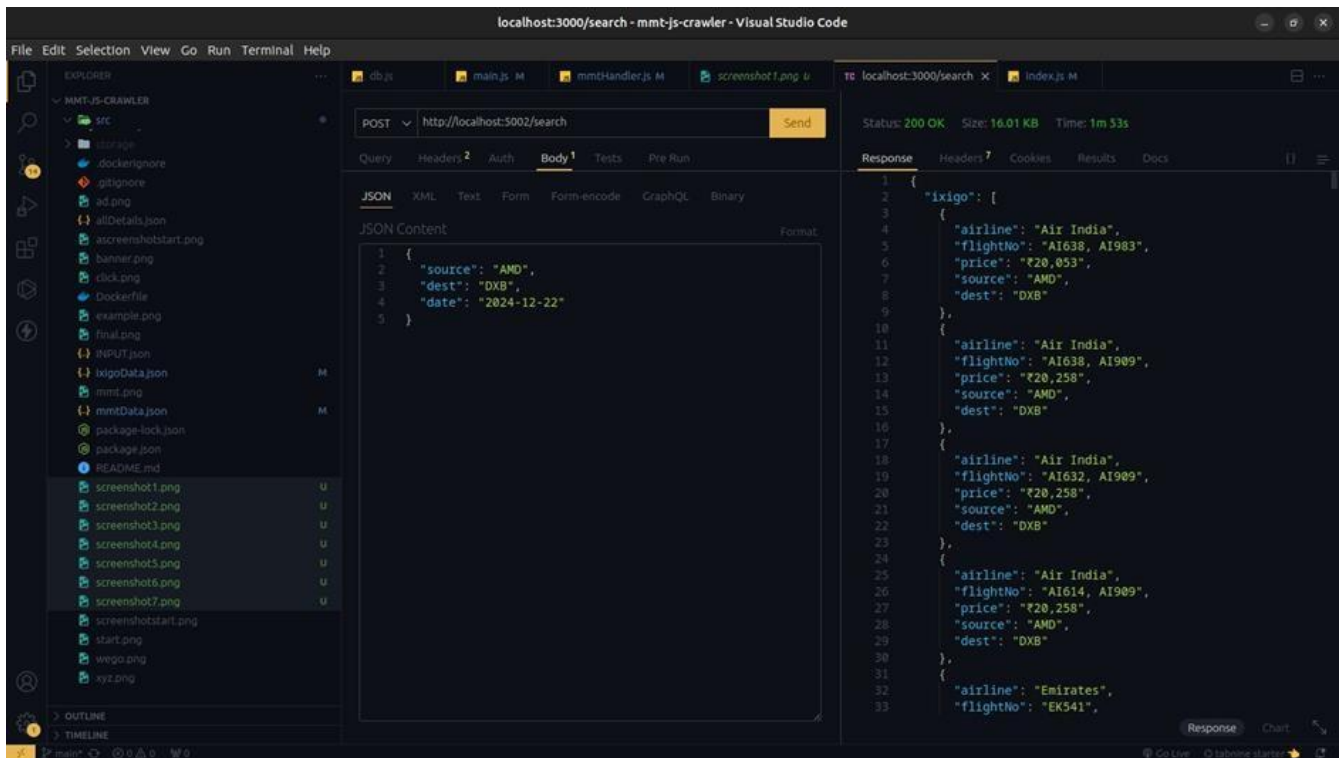
- Table structures for the data store in Mysql.



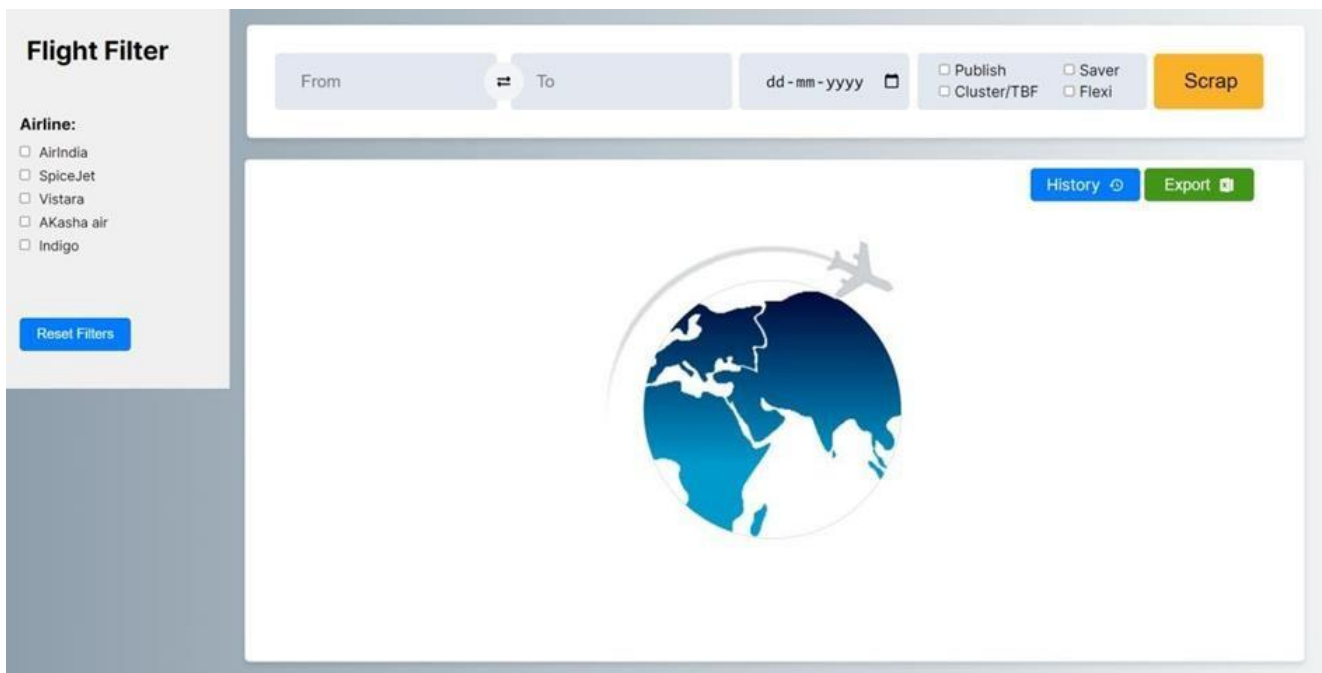


- API call for searching dynamic data on multiple sites.





- Admin Dashboard to search dynamically route and scrap data.



- Select/Enter departure Location.

Flight Filter

Airline:

☐ AirIndia

☐ SpiceJet

☐ Vistara

☐ AKasha air

☐ Indigo

Reset Filters

AMD

To

dd-mm-yyyy

☐ Publish

☐ Saver

☐ Cluster/TBF

☐ Flexi

Scrap

AMD

Ahmedabad

Sardar Vallabh Bhai Patel International Airport

BPX

Bangda

Qamdo Bangda Airport


SXX

Saumlaki-Yamdena Island

Mathilda Batlayeri Airport

History

Export



- Select Departure Date for search.

Flight Filter

Airline:

☐ AirIndia

☐ SpiceJet

☐ Vistara

☐ AKasha air

☐ Indigo

Reset Filters

AMD

To

dd-mm-yyyy

☐ Publish

☐ Saver

☐ Cluster/TBF

☐ Flexi

Scrap

History

Export

May, 2024


↑

↓

Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Clear

Today



- Scrapped data and the comparison of multiple sites and calculating markup.

Flight Filter

Airline:

☐ AirIndia
☐ SpiceJet
☐ Vistara
☐ AKasha air
☐ Indigo

Reset Filters

From

To

dd/mm/yyyy

☐ Publish
☐ Cluster/TBF

☐ Saver
☐ Flexi

Scrap

History

Export

Departure Location	Departure Date	Arrival Location	Arrival Date	Airline Name	Direct	Flight No.	FairType	TBO Publis
Ahmedabad (AMD)	29 May 2024	New Delhi (DEL)	29 May 2024	Akasa Air	1	1145	Saver	4707
Ahmedabad (AMD)	29 May 2024	New Delhi (DEL)	29 May 2024	SpiceJet	1	8194	Publish	6324
Ahmedabad (AMD)	29 May 2024	New Delhi (DEL)	29 May 2024	Vistara	1	966	Publish	4900
Ahmedabad (AMD)	29 May 2024	New Delhi (DEL)	29 May 2024	Vistara	1	976	Publish	5215
Ahmedabad (AMD)	29 May 2024	New Delhi (DEL)	29 May 2024	Vistara	1	906	Publish	5215
Ahmedabad (AMD)	29 May 2024	New Delhi (DEL)	29 May 2024	Indigo	1	2736	Saver	5232
Ahmedabad (AMD)	29 May 2024	New Delhi (DEL)	29 May 2024	Indigo	1	2209	Saver	5286
Ahmedabad (AMD)	29 May 2024	New Delhi (DEL)	29 May 2024	Indigo	1	2047	Saver	5286
Ahmedabad (AMD)	29 May 2024	New Delhi (DEL)	29 May 2024	Indigo	1	2032	Saver	5286

CHAPTER 8: CONCLUSION AND FUTURE WORK

CHAPTER 8: CONCLUSION AND FUTURE WORK

8.1 Conclusion

- ✓ In conclusion, the flight data scraping project presents a promising solution for streamlining the process of gathering and analyzing flight information. By centralizing data from various online platforms, the project aims to provide users with a convenient and comprehensive view of available flights, including pricing details and route options. The primary objective of simplifying the search for the best flight deals through an easy-to-use dashboard aligns with the needs of modern travelers. Future developments, such as expanding data sources and incorporating advanced analysis techniques, promise to further enhance the system's capabilities and user experience.
- ✓ Overall, the project endeavors to evolve into a robust platform that empowers travelers with efficient and informed solutions for booking flights. By leveraging technology and building upon existing methodologies, the project contributes to the advancement of travel technology and enhances the travel booking experience for users.

8.2 Future work

Future work for the travel rate scraping project includes expanding data sources, integrating advanced data analysis techniques, and incorporating user preferences for personalized recommendations. Real-time updates and mobile application development are suggested to enhance accessibility, while partnerships with travel agencies and airlines can improve service quality and access to exclusive deals. User feedback mechanisms will drive iterative improvements, ensuring the system remains responsive to evolving user needs and industry trends. Through these enhancements, the system can evolve into a robust platform, providing travelers with efficient and informed flight booking solutions.

CHAPTER 9: REFERENCES

CHAPTER 9: REFERENCES

- <https://www.w3schools.in/sdlc/agile-model>
- <https://nextjs.org/docs>
- <https://www.youtube.com>
- <https://nodejs.org/docs/latest/api/documentation.html>