VIETNAM NATIONAL UNIVERSITY – HOCHIMINH CITY
THE INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DATA SCIENCE AND DATA VISUALIZATION
IT138IU

FINAL REPORT

Viet Nam Aviation Analytics

By Group 04 – Member List

1	Nguyen Minh Viet	ITDSIU21130	Team Member
2	Le Huynh Nha Nguyen	ITDSIU21058	Team Member
3	Phan Bao Tran	ITDSIU21125	Team Member

Instructor: Dr. Tran Thanh Tung

ACKNOWLEGEMENT

The team would like to express our deepest gratitude to Dr. Tran Thanh Tung for his invaluable guidance, support, and expertise throughout the course of this project. His insightful feedback, encouragement, and thoughtful suggestions were crucial in shaping the direction of our work. The team are truly grateful for the opportunity to work under his mentorship and for the knowledge the team has gained from his expertise.

In addition to the specialized knowledge imparted during the lectures, Dr. Tran Thanh Tung for the invaluable practical experiences he shared throughout both the theoretical and practical sessions. These insights have greatly helped us apply the knowledge effectively to our project, enriching our ability to translate theory into real-world practice.

Due to the limitations in knowledge and time, the project may still have some shortcomings. The team looks forward to receiving feedback from Dr. Tung to further improve and refine our work. The team would also like to express our heartfelt gratitude for the support and valuable contributions provided by Dr. Tung throughout our learning process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In the ever-evolving aviation industry, the role of data analysis and visualization has become increasingly indispensable. This study provides profound insights into how these powerful tools can help customers and stakeholders harness aviation data to make optimal decisions. Through a modern, interactive dashboard, users are not only presented with raw numbers but are also able to uncover the hidden narratives behind the data - covering ticket prices, flight volumes, peak hours, and connectivity across regions.

Notably, the dashboard empowers users to filter information by date, diving into historical data to analyze and predict future trends. This enables customers to easily access detailed insights, such as ticket prices from various airlines on specific dates, the number of flights operating, peak time slots, and the volume of routes departing from their local airport to other provinces. With this data at hand, users can identify ideal destinations, compare costs and schedules, and make decisions tailored to their personal needs and preferences.

The dashboard is not merely an analytical tool but also serves as a strategic guide for travelers looking to enhance their planning and efficiency. Combining historical data with predictive capabilities creates opportunities for users to save costs, plan effectively, and maximize convenience. Moreover, this information holds strategic value for airlines and managers, enabling them to tap into market potential, adjust operations, and better meet consumer demands.

In summary, this study illustrates the transformative power of data and visualization in connecting individuals with innovative and optimized solutions, thereby fostering sustainable and comprehensive growth within the aviation sector.

Chapter I

INTRODUCTION

1. Overview

The AEVION provides an interactive and user-friendly platform for analyzing key aviation metrics such as flight volumes, passenger trends, pricing patterns, and route coverage. By incorporating intuitive navigation and insightful visualizations, the application enables users to explore aviation data and gain actionable insights tailored to their needs. Users can filter data by airline, departure and arrival locations, and specific dates, allowing for personalized and detailed exploration of historical trends or future predictions.

The dashboard features various visualization tools, including a bar chart comparing total flights by airline, a heatmap showcasing flight density by hour, a geographic map of passenger routes and flight coverage, and a scatter plot analyzing daily pricing trends. These visualizations are further complemented by summary metrics like total travelers and flights, offering a high-level overview of industry activity.

The source code can be found on GitHub via this link.

2. Goal

The core objective of the Vietnam Aviation Analytics dashboard is to serve as a reliable companion, providing customers with comprehensive information to make intelligent, cost-effective, and efficient travel decisions. With a strong focus on enhancing user experience, the dashboard enables customers to seamlessly access and analyze detailed

data on ticket prices, flight schedules, route coverage, and especially peak travel hours. Through flexible filters for airlines, departure points, destinations, and specific dates, users can explore ticket price trends, identify peak travel times, and understand the volume of flights operating during various time slots. This empowers customers to compare travel options, choose optimal schedules, and manage their time and budget more effectively.

The dashboard goes beyond merely presenting data by integrating powerful visualization tools such as bar charts showing total flights by airline, heatmaps illustrating flight density by time slots, and scatter plots for tracking daily ticket price trends. Particularly, detailed insights into peak travel hours allow customers to identify periods of high passenger traffic, enabling them to plan better and avoid congestion if necessary. Additionally, the ability to analyze historical data helps customers forecast future trends, allowing them to plan their trips strategically and effectively.

In summary, the Vietnam Aviation Analytics dashboard is not just a data analysis tool but a comprehensive platform that supports customers in optimizing their journeys and enhancing their overall travel experience in the aviation industry. With an intuitive interface, highly interactive features, and easy access to critical information, this dashboard bridges the gap between complex aviation data and smart travel decisions, providing real value to customers.

3. Techniques and Tools Used

(i) IDE for web programming: Visual Studio Code - HTML/CSS/D3.js/React/TypeScript/Vite.

(ii) Draft the UI of the dashboard: Tableau, Python.

(iii) Cleansing data: Python Jupyter Notebook by Google Colab.

(iv) Code version management: GitHub Desktop.

Chapter II

TASK TIMELINE

Table III-1

Task Coordination

Stage	Task	Responsible	Week
Planning	Research about statistic and implementation of the Dashboard	All	6 weeks
	Collect data	Bảo Trân	
	Processing data	Nhã Nguyên	
	Agree on communication, workflow, and tools to use	All	
	Decide scopes, learning aims, and goals for the project	All	
	Specify the goals and write structure of storytelling	All	
	Specify the template of the dashboard	All	
Implementation	Complete the dashboard by using Tableau	Nhã Nguyên	4 weeks
	Implement the structure of the web	Minh Việt	
	Implement charts on the web	Bảo Trân, Nhã Nguyên	
	Implement interactivity on charts	Bảo Trân, Nhã Nguyên	
	Implement CSS features of the web	Minh Việt	
Testing	Fix and change the website dashboard to better match cases and stories	Minh Việt	2 weeks
Final Report	Write report	All	2 weeks

Chapter III

METHODOLOGY

1. Implementation Layout and Chart

The layout of this project draws inspiration from airline websites, aligning with the central theme of report articles. In the initial design, the web was planned to include an introductory animation transitioning to the main page. However, this feature was ultimately scrapped out due to the complexity of implementing it in the code. As a result, when the website first loads, users are directed immediately to the home page, bypassing any introductory sequence.

1.1. Navigation Bar

Positioned on the left side of the screen, the navigation bar serves as a central component of the interface. It displays the fictional brand logo and brand name at the top. Below these, three interactive tabs are available, enabling users to switch between the home page, the chart page, and a temporary placeholder page. Each tab incorporates an animation to enhance user interaction. The tab corresponding to the current page darkens to signify the active selection. While hovering over other tabs also causes them to darken, signaling potential navigation.

The navigation bar is equipped with an additional functionality that allows it to contract or expand upon clicking an arrow icon on the top right of the navigation bar. When contracted, the bar dynamically updates its components: the brand name is hidden, and each tab name is substituted with a representative icon. Additionally, the position of the

arrow is seamlessly synchronized with the navigation bar's transitions during the contraction and expansion process. This adaptive design enhances the user experience by providing a cleaner and more flexible interface.

1.2. Mode Switch Feature

Beyond showcasing branding name, logo, and page navigation tabs, the navigation bar features a mode switch button at its bottom that toggles between light and dark modes. Upon updating the mode, the entire website adapts to the selected mode's color scheme, making the design cohesive and uniform. Notably, the appearance of the mode switch button adapts responsively as the navigation bar contracts. Additional visual enhancements include a glowing effect on the mode switch button when hovered over and a similar effect on targeted text while in dark mode.

1.3. Main Content Area

The primary content of each page is displayed in the central portion of the screen. This area automatically resizes to expand when the navigation bar is contracted, optimizing screen space. At the bottom of each page, "Back" and "Next" buttons are included for seamless and easy navigation between pages. These buttons are also designed to darken upon hover, giving users clear visual cues in interaction.

The fictional content on the homepage aligns with the central theme of a fabricated airline travel agency, presenting engaging and relatable information. The chart tab houses the project's core feature: a dashboard that visualizes an analysis of Vietnam's flight data, fulfilling the primary objective of the project. Lastly, the team tab showcases details about

our development team, the course project, and the integrated technologies used in the site's creation. By amalgamating these features, this site layout provides an intuitive and visually appealing interface that emphasizes both functionality and end-user engagement.

1.4. Horizontal Bar Chart

Data Processing: the data is grouped by code_name (airline code), and the total number of flights for each airline is calculated using the d3.rollup function.

Figure III-1

Grouping Data to Prepare for Bar Chart

```
function drawBarChart(data) {
  const counts = d3.rollup(
    data,
    (v) => v.length,
    (d) => d.code_name
  );
  const barData = Array.from(counts, ([code_name, count]) => ({
    code_name,
    count,
  }));
```

Scales Setup: xScale is a linear scale used to represent the total number of flights. yScale is a band scale utilized for airline names.

Figure III-2

Defining Scales for Bar Chart Axes

```
const xScale = d3
  .scaleLinear()
  .domain([0, d3.max(barData, (d) => d.count)])
  .range([0, width - margin.left - margin.right]);

const yScale = d3
  .scaleBand()
  .domain(barData.map((d) => d.code_name))
  .range([0, height - margin.top - margin.bottom])
  .padding(0.1);
```

SVG and Group Creation: the chart is constructed by appending a group (<g>) to the SVG element, with appropriate margins for clear layout.

Figure III-3

Setting Up the SVG and Rendering Bars

```
bars
  .enter()
  .append("rect")
  .attr("y", (d) => yScale(d.code_name))
  .attr("height", yScale.bandwidth())
  .attr("x", 0)
  .attr("width", 0)
  .attr("fill", (d, i) => `url(#gradient-${i})`) // Apply gradient
  .on("mouseover", function (event, d) {
    tooltip.style("visibility", "visible").text(`${d.code_name}: ${d.count}`);
  })
  .on("mousemove", (event) => {
    tooltip
      .style("top", `${event.pageY + 10}px`)
      .style("left", `${event.pageX + 10}px`);
  })
  .on("mouseout", () => tooltip.style("visibility", "hidden"))
  .transition()
  .duration(1000)
  .attr("width", (d) => xScale(d.count));

chart.append("g").call(d3.axisLeft(yScale));
chart
  .append("g")
  .attr("transform", `translate(0,${height - margin.top - margin.bottom})`)
  .call(d3.axisBottom(xScale));
```

Interactive Features: tooltip displays detailed information such as the airline name and the total number of flights when hovering over the bars.

Figure III-4

Tooltip Configuration

```
// Tooltip
const tooltip = d3
  .select("body")
  .append("div")
  .attr("class", "tooltip")
  .style("position", "absolute")
  .style("visibility", "hidden")
  .style("background", "#fff")
  .style("border", "1px solid #ccc")
  .style("padding", "5px")
  .style("border-radius", "4px");
```

1.5. Scatter Plot

Data Processing: the data is processed to compute the average ticket price (avg_price) for each airline on each specific date. This is achieved by grouping the dataset by departure date (departure_date) and airline code (code_name), then calculating the mean of ticket prices for each group. The resulting data is structured to include dates, airlines, and their corresponding average ticket prices, preparing it for visualization.

Figure III-5

Grouping Data to Prepare for Scatter Plot

```
function drawScatterPlot(data) {
  const processedData = Array.from(
    d3.rollup(
      data,
      (v) => d3.mean(v, (d) => d.total_price),
      (d) => d.departure_date.split(" ")[0],
      (d) => d.code_name
    ),
    ([date, airlines]) => ({
      date: date,
      airlines: Array.from(airlines, ([code_name, avg_price]) => ({
        code_name: code_name,
        avg_price: avg_price,
      })),
    })
  );
```

Scales Setup: xScale is configured as a time scale, ensuring precise and smooth representation of the date range. The vertical axis yScale utilizes a linear scale to display ticket prices, ranging from 0 to the maximum average ticket price in the dataset.

Figure III-6

Defining Scales for Scatter Plot Axes

```
const xScale = d3
  .scaleTime()
  .domain([new Date("2021-04-01"), new Date("2021-05-04")])
  .range([0, width]);

const yScale = d3
  .scaleLinear()
  .domain([
    0,
    d3.max(processedData, (d) => d3.max(d.airlines, (a) => a.avg_price)),
  ])
  .range([height, 0]);
```

SVG and Plot Creation: an SVG element was created with carefully defined margins to ensure a clean and well-organized layout. Data points were represented as circles, with their positions determined by the respective date on the x-axis and average ticket price on the y-axis. A color scale was applied to distinguish airlines, enhancing the chart's visual clarity and appeal.

Figure III-7

Setting Up the SVG and Rendering Circles

```
processedData.forEach((day) => {
  day.airlines.forEach((airline) => {
    svg
      .append("circle")
      .attr("cx", xScale(new Date(day.date)))
      .attr("cy", yScale(airline.avg_price))
      .attr("r", 0)
      .attr("fill", colorScale(airline.code_name))
      .attr("stroke", "black")
      .attr("stroke-width", 1.5)
      .on("mouseover", function () {
        d3.select(this).attr("r", 6);
        tooltip
          .style("visibility", "visible")
          .text(`${airline.code_name}: ${airline.avg_price.toFixed(2)} VND`);
      })
      .on("mousemove", (event) => {
        tooltip
          .style("top", `${event.pageY + 10}px`)
          .style("left", `${event.pageX + 10}px`);
      })
      .on("mouseout", function () {
        d3.select(this).attr("r", 4.5);
        tooltip.style("visibility", "hidden");
      })
      .transition()
      .duration(1000)
      .attr("r", 4.5);
  });
});

svg.append("g").call(d3.axisLeft(yScale));
svg
  .append("g")
  .attr("transform", `translate(0, ${height})`)
  .call(
    d3
      .axisBottom(xScale)
      .ticks(d3.timeDay.every(2))
      .tickFormat(d3.timeFormat("%d-%b"))
  )
  .selectAll("text")
  .attr("transform", "rotate(-45)")
  .style("text-anchor", "end");
```

1.6. Heat Map

To visualize the data effectively, first the team aggregated and transformed to match the requirements of the heatmap.

Figure III-8

Data Aggregation and Transformation

```
function drawHeatmap(
  data: Array<{ code_name: string; departure_time: string }>
): void {
  const processedData: Array<[string, Array<[number, number]>]> = d3.rollups(
    data,
    (v: Array<{ code_name: string; departure_time: string }>) => v.length,
    (d: { code_name: string }) => d.code_name,
    (d: { departure_time: string }) => +d.departure_time.split(":")[0]
  );

  const heatmapData: Array<{ airline: string; hour: number; count: number }> =
    processedData.flatMap(([airline, hours]) =>
      Array.from(hours, ([hour, count]) => ({ airline, hour, count })));
```

The team set up the SVG Canvas, the visualization is rendered inside an SVG container with appropriate margins, width, and height.

Figure III-9

Setting Up the SVG Canvas

```
const margin: { top: number; right: number; bottom: number; left: number } =
  { top: 30, right: 50, bottom: 30, left: 90 };
const width: number = 530 - margin.left - margin.right;
const height: number = 200 - margin.top - margin.bottom;

const svg: d3.Selection<SVGSVGElement, unknown, null, undefined> = d3
  .select("#heatmap")
  .html("")
  .append("svg")
  .attr("width", width + margin.left + margin.right)
  .attr("height", height + margin.top + margin.bottom)
  .append("g")
  .attr("transform", `translate(${margin.left},${margin.top})`);
```

Next step, scales are used to map data values to visual positions and color intensities.

Figure III-10

Defining Scales

```
const maxCount: number = d3.max(heatmapData, (d) => d.count) || 0;
const colorScale: d3.ScaleSequential<string> = d3
  .scaleSequential(d3.interpolateBlues)
  .domain([0, maxCount]);
```

Next step, the team added a tooltip to enhance interactivity and provide detailed information.

Figure III-11

Tooltip Configuration

```
const tooltip = d3
  .select("body")
  .append("div")
  .attr("class", "tooltip")
  .style("position", "absolute")
  .style("opacity", 0)
  .style("background", "#fff")
  .style("color", "#000")
  .style("border", "1px solid #ccc")
  .style("padding", "10px")
  .style("box-shadow", "0px 0px 6px rgba(0,0,0,0.1)")
  .style("border-radius", "4px");
```

Then, rectangles were drawn for each airline-hour combination, with dynamic color and hover effects, including a bold outline and tooltip display.

Figure III-12

Rendering Heatmap Cells

```
svg
  .selectAll("rect")
  .data(heatmapData)
  .enter()
  .append("rect")
  .attr("x", (d) => xScale(d.hour))
  .attr("y", (d) => yScale(d.airline))
  .attr("width", xScale.bandwidth())
  .attr("height", yScale.bandwidth())
  .style("fill", (d) => colorScale(d.count || 0))
  .style("stroke", "none")
  .on("mouseover", function (event, d) {
    tooltip.transition().duration(200).style("opacity", 1);
    tooltip
      .html(
        `<strong>Airline:</strong> ${d.airline}<br>
        <strong>Hour:</strong> ${d.hour}<br>
        <strong>Flights:</strong> ${d.count}`
      )
      .style("left", `${event.pageX + 10}px`)
      .style("top", `${event.pageY - 10}px`);
    d3.select(this).style("stroke", "black").style("stroke-width", 2);
  })
  .on("mousemove", (event) => {
    tooltip
      .style("left", `${event.pageX + 10}px`)
      .style("top", `${event.pageY - 10}px`);
  })
  .on("mouseout", function () {
```

Finally, the team added X and Y axes to label hours and airlines, with rotated X-axis labels for readability.

Figure III-13

Add Axes

```
  svg.append("g").call(d3.axisLeft(yScale));
  svg
    .append("g")
    .attr("transform", `translate(0, ${height})`)
    .call(d3.axisBottom(xScale).tickFormat((d) => `${d}:00`))
    .selectAll("text")
    .attr("transform", "rotate(-45)")
    .attr("x", -10);
}
```

1.7. Map

The map of Vietnam is rendered using a Mercator projection centered and scaled to fit Vietnam's geographic region. The map is drawn using GeoJSON data and styled with colors and strokes.

Figure III-14

Setting Up the Map

```
const projection = d3
  .geoMercator()
  .center([107, 16]) // Center coordinates for Vietnam
  .scale(2000) // Zoom level
  .translate([mapWidth / 2, mapHeight / 2]);

const path = d3.geoPath().projection(projection);

const mapSvg = d3
  .select<SVGSVGElement, unknown>("#vietnam-map")
  .attr("width", mapWidth)
  .attr("height", mapHeight);
```

Next, each province is highlighted on hover, and a tooltip is displayed with the province name.

Figure III-15

Adding Interactivity to the Map

```
mapSvg
.selectAll<SVGPathElement, VietnamFeature>("path")
.data(vietnam.features)
.enter()
.append("path")
.attr("d", path)
.attr("fill", "#ffb733")
.attr("stroke", "#fff")
.attr("stroke-width", 1.5)
.on("mouseover", function (_event: MouseEvent, d: VietnamFeature) {
  d3.select(this).transition().duration(200).attr("fill", "#ffd166");
  tooltip
    .style("visibility", "visible")
    .html(`<strong>Province:</strong> ${d.properties.name}`);})
.on("mousemove", (event: MouseEvent) => {
  tooltip
    .style("top", `${event.pageY + 10}px`)
    .style("left", `${event.pageX + 10}px`);})
.on("mouseout", function () {
  d3.select(this).transition().duration(200).attr("fill", "#ffb733");
  tooltip.style("visibility", "hidden");});

drawFlightRoutesFromData(mapSvg, flightData);
```

Flight routes are drawn as animated lines connecting the origin and destination coordinates. The lines transition smoothly, visually representing flight paths.

Figure III-16

Drawing Flight Routes

```
mapSvg
  .append("line")
  .attr("x1", fromCoords[0])
  .attr("y1", fromCoords[1])
  .attr("x2", fromCoords[0]) // Start from origin
  .attr("y2", fromCoords[1])
  .attr("stroke", "blue")
  .attr("stroke-width", 2)
  .transition()
  .duration(1000)
  .attr("x2", toCoords[0]) // Transition to destination
  .attr("y2", toCoords[1])
```

Origins are marked with red circles, while destinations are highlighted using custom pin-shaped icons.

Figure III-17

Marking Flight Origins and Destinations

```
// Add "from" marker as red circle
mapSvg
  .append("circle")
  .attr("cx", fromCoords[0])
  .attr("cy", fromCoords[1])
  .attr("r", 5)
  .attr("fill", "red")

// Add "to" marker as custom pin icon with transition
const destinationMarker = mapSvg
  .append("use")
  .attr("xlink:href", "#destination-icon")
  .attr("x", fromCoords[0] - 12) // Start at origin
  .attr("y", fromCoords[1] - 24) // Start at origin
  .transition()
  .duration(1000)
  .attr("x", toCoords[0] - 12)
  .attr("y", toCoords[1] - 24)
```

Then, the team added interactive tooltips displaying detailed flight information, such as flight codes, cities, and airports.

Figure III-18

Adding Tooltips for Routes

```
.on("mouseover", () => {
  tooltip.style("visibility", "visible").html(`
    <strong>Flight Code:</strong> ${flight.code_name}<br>
    <strong>From:</strong> ${flight.airport_from} (${flight.from})<br>
    <strong>To:</strong> ${flight.airport_to} (${flight.to})
  `);
})
```

The team created updateFilteredFlightRoutes() function, which dropdown menus allow users to filter flight routes based on the selected origin and destination cities.

Figure III-19

Filtering Flight Routes

```
function updateFilteredFlightRoutes(mapSvg, flightData) {
  const selectedFrom = d3.select("#from").node().value;
  const selectedTo = d3.select("#to").node().value;

  const filteredData = flightData.filter(
    (d) =>
      (selectedFrom === "All" || d.from === selectedFrom) &&
      (selectedTo === "All" || d.to === selectedTo)
  );

  drawFlightRoutesFromData(mapSvg, filteredData);
}
```

2. Dashboard Journey

Figure III-20

Dashboard

The Vietnam Aviation Analytics dashboard is a reliable tool that helps customers make smart and cost-effective flight decisions by providing detailed information on ticket prices, flight schedules, route coverage, and peak travel hours. For instance, travelers can use the dashboard to compare ticket prices across airlines, identify peak travel hours, and analyze pricing trends, such as whether ticket prices are higher at the beginning or the end of the month. This allows users to make well-informed and practical choices. With interactive charts and flexible filters, users can effortlessly compare options and select optimal schedules to effectively manage their time and budget.

The dashboard is structured around four key components to optimize usability. First, data filters allow users to refine their search based on airline, departure point, destination,

and specific dates within a month. Second, interactive charts, including bar charts, heatmaps, and scatter plots, provide visual analysis of flight density, ticket price trends, and airline performance. Third, a route map visualizes flight paths and frequencies, giving users a comprehensive overview of Vietnam's aviation network. Lastly, summary statistics display total flight numbers and passenger counts, offering a quick snapshot of overall activity.

One of the dashboard's standout features is its flexible filtering system, which enables users to customize their data views. By adjusting parameters such as airline, departure location, destination, and dates, users can tailor the displayed information to their specific needs. This flexibility ensures that the dashboard caters to a wide range of travel planning requirements.

Interacting with the dashboard is straightforward and intuitive. Users begin by selecting their preferred filtering criteria, such as airline, departure and destination points, or specific dates, to refine the displayed data. They can then click on various chart elements to access detailed information, such as ticket prices or the number of flights during specific time periods. This seamless interaction allows users to explore data efficiently and make well-informed decisions about their travel plans.

3. Cleaning Data

Before analyzing and visualizing the data in the Vietnam Aviation Analytics dashboard, it is essential to ensure the dataset's quality and consistency. Raw data often contains errors, missing values, and inconsistencies that can impact the accuracy and

reliability of insights. Therefore, the data cleaning process was carried out to prepare the dataset, making it ready for effective analysis and visualization. Below are the steps taken during this process, along with the tools and methods used.

3.1. Initial Data Inspection

The dataset was loaded and reviewed for structure, column names, and key statistics to understand its content. Checks were performed for missing values and duplicate entries to ensure data completeness and consistency, providing a solid foundation for further processing.

Figure III-21

Load Data

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import numpy as np
import pandas as pd
```

```
df = pd.read_csv('/content/flight_.csv')
df.head()
```

	id	code	code_name	f_code	f_price	fees	total_price	from	airport_from	to	airport_to	type	departure_date	departure_time	arrival_date	arrival_time	flight_durat
0	vj16169846100	vj	Vietjet	VJ122	199000	650900	849900	TP HCM	Tân Sơn Nhất (SGN)	Hà Nội	Nội Bài (HAN)	Eco	2021-04-01	04:50:00	2021-04-01	07:10:00	
1	vna16169846200	vna	Vietnam Airlines	VN250	299000	670000	969000	TP HCM	Tân Sơn Nhất (SGN)	Hà Nội	Nội Bài (HAN)	Economy (EL)-P	2021-04-01	05:00:00	2021-04-01	07:10:00	
2	vj16169846231	vj	Vietjet	VJ176	199000	650900	849900	TP HCM	Tân Sơn Nhất (SGN)	Hà Nội	Nội Bài (HAN)	Eco	2021-04-01	05:25:00	2021-04-01	07:35:00	
3	qh16169846270	qh	Bamboo Airways	QH202	399000	672000	1071000	TP HCM	Tân Sơn Nhất (SGN)	Hà Nội	Nội Bài (HAN)	Eco Saver	2021-04-01	05:45:00	2021-04-01	07:55:00	
4	vj16169846302	vj	Vietjet	VJ120	199000	650900	849900	TP HCM	Tân Sơn Nhất (SGN)	Hà Nội	Nội Bài (HAN)	Eco	2021-04-01	06:00:00	2021-04-01	08:10:00	

Figure III-22

Check Null Value

```
missing_values = df.isnull().sum()
missing_values
```

	0
id	0
code	0
code_name	0
f_code	0
f_price	0
fees	0
total_price	0
from	0
airport_from	0
to	0
airport_to	0
type	0
departure_date	0
departure_time	0
arrival_date	0
arrival_time	0
flight_duration_hours	0
Latitude	0
Longitude	0

dtype: int64

Figure III-23

Describe the Data

```
dataset_description = df.describe(include='all')
dataset_description
```

	id	code	code_name	f_code	f_price	fees	total_price	from	airport_from	to	airport_to	type	departure_date	departure_time	arrival_date	arrival_time	flight_duration_hours	Latitude	Longitude
count	19846	19846	19846	19846	1.984600e+04	1.984600e+04	1.984600e+04	19846	19846	19846	19846	19846	19846	19846	19846	19846	19846.000000	19846.000000	19846.000000
unique	19846	4	5	722	NaN	NaN	NaN	13	13	13	13	42	34	224	35	222	NaN	NaN	NaN
top	vj16169846100	vna	Vietjet	VN6050	NaN	NaN	NaN	TP HCM	Tân Sơn Nhất (SGN)	TP HCM	Tân Sơn Nhất (SGN)	Eco	2021-05-03	09:35:00	2021-05-03	09:15:00	NaN	NaN	NaN
freq	1	9233	7423	39	NaN	NaN	NaN	7241	7241	7294	7294	7348	614	221	614	268	NaN	NaN	NaN
mean	NaN	NaN	NaN	NaN	4.121952e+05	6.756739e+05	1.087869e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.170804	14.756273	106.608776
std	NaN	NaN	NaN	NaN	5.937399e+05	5.264527e+04	6.456091e+05	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	3.562191	4.426578	1.227761
min	NaN	NaN	NaN	NaN	2.600000e+04	6.419000e+05	6.806000e+05	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-22.750000	10.045200	103.984000
25%	NaN	NaN	NaN	NaN	9.900000e+04	6.510000e+05	7.499000e+05	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.330000	10.823100	105.854200
50%	NaN	NaN	NaN	NaN	1.990000e+05	6.600000e+05	8.590000e+05	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.750000	12.238800	106.629700
75%	NaN	NaN	NaN	NaN	4.690000e+05	6.710000e+05	1.139000e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.080000	20.844900	106.688100
max	NaN	NaN	NaN	NaN	5.309000e+06	1.152000e+06	6.460000e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.750000	21.028500	109.219800

3.2. Data Cleaning

The data cleaning process involved identifying and correcting anomalies in ticket prices by comparing the total price with the sum of the base price and fees, making necessary

adjustments to the base price column. Date and time information were standardized by splitting the departure and arrival timestamps into separate columns for date, time, month, and year, while flight durations were calculated in both hours and minutes. Invalid date and time entries, such as negative flight durations, were resolved by appropriately adjusting the arrival dates. Fortunately, the dataset we collected contained no null values or duplicate rows, which streamlined the cleaning and analysis process. These steps ensured the accuracy and consistency of the data for reliable analysis.

Figure III-24

Identifying and Correcting Anomalies in Ticket Prices

```
# Identify potential anomalies
df['price_anomaly'] = df['f_price'] + df['fees'] != df['total_price']

# Filter rows with anomalies (if necessary)
anomalies = df[df['price_anomaly']]
print("Anomalies found:\n", anomalies)
```

Figure III-25

Identifying Rows with Incorrect Departure and Arrival Times

```
# Calculate flight time (hours)
df['flight_duration_hours'] = (df['f_time_to'] - df['f_time_from']).dt.total_seconds() / 3600

# Filter rows with negative values in the flight_duration_hours column
negative_duration_rows = df[df['flight_duration_hours'] < 0]
print(negative_duration_rows)
```

```
       flight_duration_hours  Latitude  Longitude  price_anomaly
55                    -21.83   10.8231   106.6297          False
56                    -21.83   10.8231   106.6297          False
58                    -21.83   10.8231   106.6297          False
59                    -21.83   10.8231   106.6297          False
114                   -21.83   21.0285   105.8542          False
...                      ...       ...        ...            ...
19641                 -22.17   18.6763   105.6813          False
19670                 -22.58   16.4637   107.5909          False
19711                 -22.58   16.0471   108.2068          False
19712                 -22.67   16.0471   108.2068          False
19713                 -22.67   16.0471   108.2068          False

[444 rows x 20 columns]
```

Figure III-26

Standardizing Date and Time Information

```
# Convert datetime columns to datetime format
df['departure_date'] = pd.to_datetime(df['departure_date'])
df['arrival_date'] = pd.to_datetime(df['arrival_date'])


df['departure_datetime'] = df['departure_date'] + pd.to_timedelta(df['departure_time'])
df['arrival_datetime'] = df['arrival_date'] + pd.to_timedelta(df['arrival_time'])

# Show recalculated values
print(df[['departure_datetime', 'arrival_datetime', 'flight_duration_hours']])
```

```
       departure_datetime    arrival_datetime  flight_duration_hours
0     2021-04-01 04:50:00 2021-04-01 07:10:00               2.333333
1     2021-04-01 05:00:00 2021-04-01 07:10:00               2.166667
2     2021-04-01 05:25:00 2021-04-01 07:35:00               2.166667
3     2021-04-01 05:45:00 2021-04-01 07:55:00               2.166667
4     2021-04-01 06:00:00 2021-04-01 08:10:00               2.166667
...                   ...                 ...                    ...
19841 2021-05-04 14:45:00 2021-05-04 16:25:00               1.666667
19842 2021-05-04 15:10:00 2021-05-04 16:50:00               1.666667
19843 2021-05-04 16:30:00 2021-05-04 18:10:00               1.666667
19844 2021-05-04 17:45:00 2021-05-04 19:25:00               1.666667
19845 2021-05-04 18:35:00 2021-05-04 20:15:00               1.666667

[19846 rows x 3 columns]
```

3.3. Geographic Coordinates Standardization

The process of geographic standardization involved assigning coordinates to airports by mapping departure (from) and destination (to) locations to their corresponding latitude and longitude values. New columns for latitude and longitude were added to the dataset to provide precise geographic details. Additionally, inconsistent location names were standardized to ensure uniformity. These steps enhanced the dataset's consistency and ensured accurate geographic representation for further analysis.

Figure III-27

Geographic Coordinates

```
coordinates = {
    "TP HCM": {"latitude": 10.7769, "longitude": 106.7009},
    "Hà Nội": {"latitude": 21.0285, "longitude": 105.8542},
    "Hải Phòng": {"latitude": 20.8449, "longitude": 106.6881},
    "Đà Nẵng": {"latitude": 16.0471, "longitude": 108.2068},
    "Phú Quốc": {"latitude": 10.227, "longitude": 103.9637},
    "Nha Trang": {"latitude": 12.2388, "longitude": 109.1967},
    "Đà Lạt": {"latitude": 11.9404, "longitude": 108.4583},
    "Thanh Hóa": {"latitude": 19.8067, "longitude": 105.7852},
    "Vinh": {"latitude": 18.6796, "longitude": 105.6811},
    "Huế": {"latitude": 16.4637, "longitude": 107.5909},
    "Quảng Nam": {"latitude": 15.5729, "longitude": 108.4748},
    "Cần Thơ": {"latitude": 10.0452, "longitude": 105.7469},
    "Quy Nhơn": {"latitude": 13.7821, "longitude": 109.2198}
}
```

Figure III-28

Geographic Coordinates Standardization

```
# Add origin and destination coordinate column
df['origin_latitude'] = df['from'].map(lambda x: coordinates[x]['latitude'])
df['origin_longitude'] = df['from'].map(lambda x: coordinates[x]['longitude'])


df['destination_latitude'] = df['to'].map(lambda x: coordinates[x]['latitude'])
df['destination_longitude'] = df['to'].map(lambda x: coordinates[x]['longitude'])
```

```
print(df[['from', 'to', 'origin_latitude', 'origin_longitude', 'destination_latitude', 'destination_longitude']].head())
```

```
     from      to  origin_latitude  origin_longitude  destination_latitude  \
0  TP HCM  Hà Nội          10.7769          106.7009               21.0285
1  TP HCM  Hà Nội          10.7769          106.7009               21.0285
2  TP HCM  Hà Nội          10.7769          106.7009               21.0285
3  TP HCM  Hà Nội          10.7769          106.7009               21.0285
4  TP HCM  Hà Nội          10.7769          106.7009               21.0285

   destination_longitude
0              105.8542
1              105.8542
2              105.8542
3              105.8542
4              105.8542
```

3.4. Exporting Cleaned Data

The cleaned dataset was saved as a CSV file named **processed_flight.csv.** The exported file was thoroughly reviewed to ensure it was correctly formatted and ready for

analysis. This step ensured that the cleaned and processed data could be seamlessly utilized for further insights and visualization.

Figure III-29

Exporting Cleaned Data

```
# Save the processed file
df.to_csv('processed_flight.csv', index=False)

# Download the file from Colab
from google.colab import files
files.download('processed_flight.csv')
```

Figure III-30

Dataset after Processing

id	code	code_name	f_code	f_price	fees	total_price	from	airport_from	to	airport_to	type	departure_date	departure_time	arrival_date	arrival_time	flight_duration_hours	origin_latitude	origin_longitude	destination_latitude	destination_longitude
vj16169846100	vj	Vietjet	VJ122	199000	650900	849900	TP HCM	Tân Sơn Nhất (SGN)	Hà Nội	Nội Bài (HAN)	Eco	2021-04-01	04:50:00	2021-04-01	07:10:00	2.3333333333333335	10.7769	106.7009	21.0285	105.8542
vna16169846200	vna	Vietnam Airlines	VN250	299000	670000	969000	TP HCM	Tân Sơn Nhất (SGN)	Hà Nội	Nội Bài (HAN)	Economy (EL)-P	2021-04-01	05:00:00	2021-04-01	07:10:00	2.1666666666666665	10.7769	106.7009	21.0285	105.8542
vj16169846231	vj	Vietjet	VJ176	199000	650900	849900	TP HCM	Tân Sơn Nhất (SGN)	Hà Nội	Nội Bài (HAN)	Eco	2021-04-01	05:25:00	2021-04-01	07:35:00	2.1666666666666665	10.7769	106.7009	21.0285	105.8542
qh16169846270	qh	Bamboo Airways	QH202	399000	672000	1071000	TP HCM	Tân Sơn Nhất (SGN)	Hà Nội	Nội Bài (HAN)	Eco Saver	2021-04-01	05:45:00	2021-04-01	07:55:00	2.1666666666666665	10.7769	106.7009	21.0285	105.8542
vj16169846302	vj	Vietjet	VJ120	199000	650900	849900	TP HCM	Tân Sơn Nhất (SGN)	Hà Nội	Nội Bài (HAN)	Eco	2021-04-01	06:00:00	2021-04-01	08:10:00	2.1666666666666665	10.7769	106.7009	21.0285	105.8542
vna16169846341	vna	Vietnam Airlines	VN206	299000	670000	969000	TP HCM	Tân Sơn Nhất (SGN)	Hà Nội	Nội Bài (HAN)	Economy (EL)-P	2021-04-01	06:00:00	2021-04-01	08:25:00	2.4166666666666665	10.7769	106.7009	21.0285	105.8542
vj16169846383	vj	Vietjet	VJ132	199000	650900	849900	TP HCM	Tân Sơn Nhất (SGN)	Hà Nội	Nội Bài (HAN)	Eco	2021-04-01	06:05:00	2021-04-01	08:05:00	2.0	10.7769	106.7009	21.0285	105.8542
vna16169846412	vna	Pacific Airlines	VN6002	199000	660000	859000	TP HCM	Tân Sơn Nhất (SGN)	Hà Nội	Nội Bài (HAN)	Economy (EL)-P	2021-04-01	06:30:00	2021-04-01	09:05:00	2.5833333333333335	10.7769	106.7009	21.0285	105.8542
vj16169846454	vj	Vietjet	VJ134	409000	661900	1070900	TP HCM	Tân Sơn Nhất (SGN)	Hà Nội	Nội Bài (HAN)	Eco	2021-04-01	07:00:00	2021-04-01	09:10:00	2.1666666666666665	10.7769	106.7009	21.0285	105.8542

4. Chart Usage

4.1. Filter

The filtering feature of the dashboard allows users to customize their data view based on specific criteria. Users can filter by Airline to narrow down data for a particular carrier, or use the From and To fields to specify departure and arrival cities, enabling an analysis of routes between selected locations. Additionally, the Date filter enables users to focus on flights operating on a particular day, making it easy to analyze schedules and trends for a specific timeframe. This intuitive and flexible filtering system ensures that users can quickly access the most relevant information tailored to their needs.

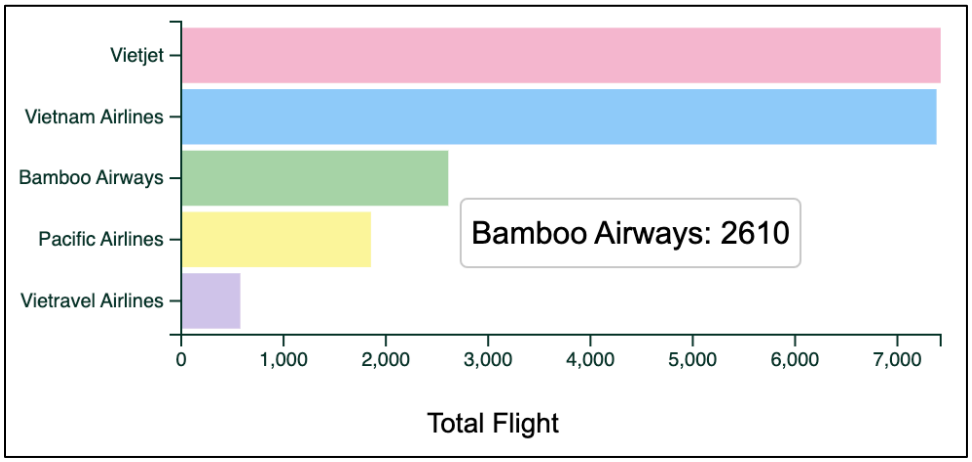
Figure III-31

Filter Bar

4.2. Horizontal Bar

Figure III-32

Horizontal Bar



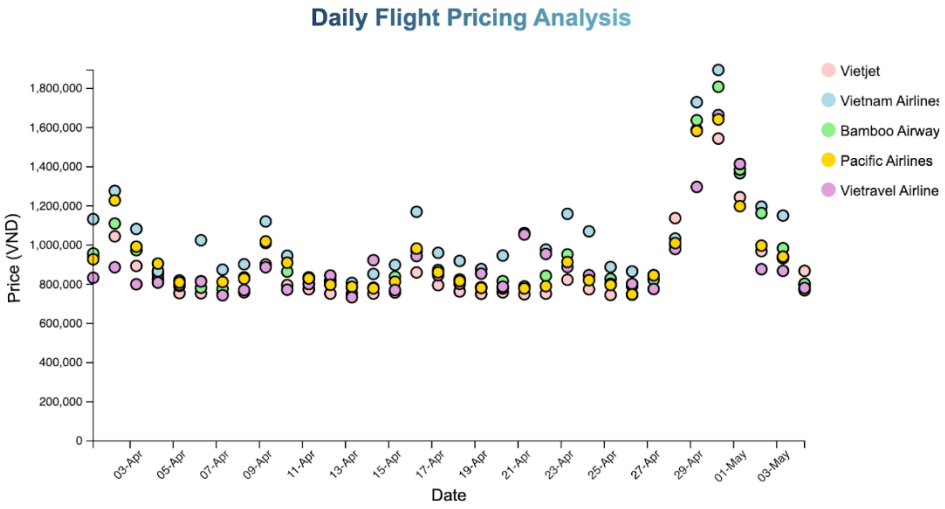
The reason why a bar chart was chosen is because it is an effective visualization method for comparing the number of flights across different airlines. Each bar represents the total number of flights for a specific airline, enabling users to quickly identify which airlines have the most or the fewest flights. This is particularly useful for analyzing the popularity of each airline, determining which airlines attract more passengers, and providing an overview of their operational performance.

The bar chart allows users to easily compare and understand flight data, making it an essential tool for analyzing trends in airline operations. For example, longer bars indicate airlines with a higher number of flights, suggesting these are more popular or have higher flight frequencies. The chart not only provides a clear visual analysis of trends but also supports decision-making, such as selecting an airline based on flight availability or popularity.

4.3. Scatter Plot

Figure III-33

Scatter Plot



The bar chart allows users to easily compare and understand flight data, making it an essential tool for analyzing trends in airline operations. For example, longer bars indicate airlines with a higher number of flights, suggesting these are more popular or have higher flight frequencies. The chart not only provides a clear visual analysis of trends but also

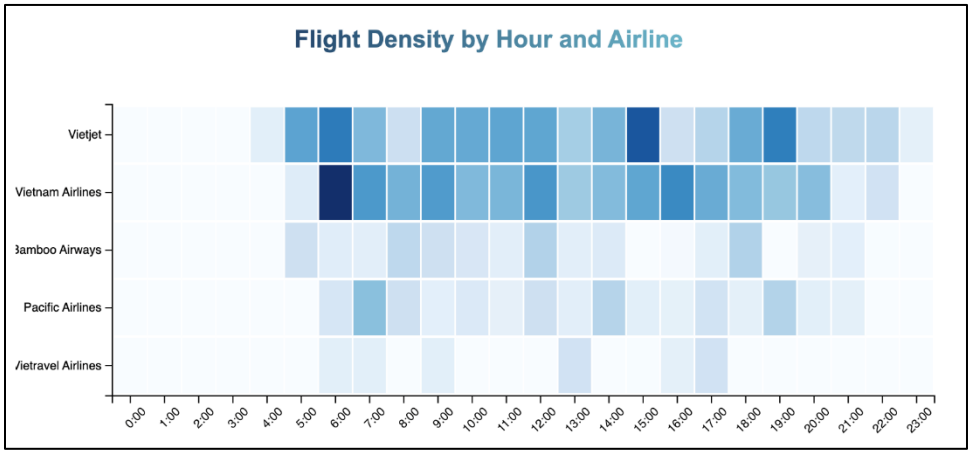
supports decision-making, such as selecting an airline based on flight availability or popularity.

The scatter plot enables users to track ticket price variations by day and compare prices across different airlines. This allows users to make budget-conscious decisions, such as selecting the airline with the most affordable ticket prices during their desired travel period. Additionally, the chart provides valuable insights into historical data, helping users optimize their travel budget and schedule.

4.4. Heat Map

Figure III-34

Heat Map



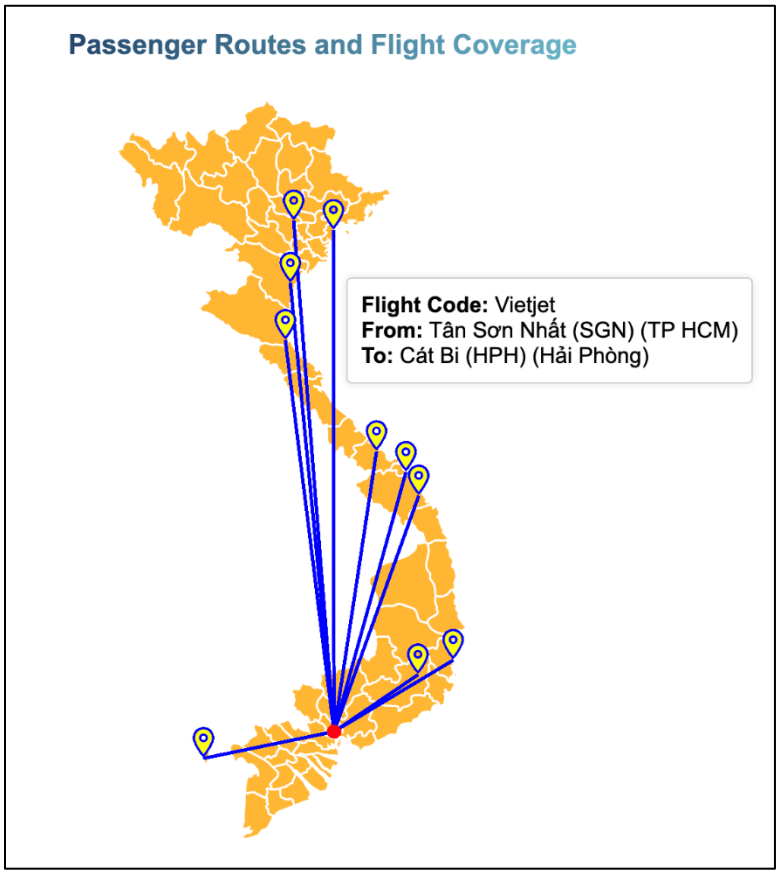
The heatmap was chosen as an effective tool to visualize flight density by hour and airline. By representing data through color intensity, this chart allows users to easily identify peak flight hours and compare activity levels across airlines during specific time frames.

The heatmap highlights peak flight hours by displaying data as color-coded cells, where darker shades represent higher flight densities. For example, darker cells indicate time periods with more flights, helping users recognize busy hours for each airline. This visualization supports users in planning their travel more effectively, either by avoiding peak hours or selecting time slots that align with their preferences.

4.5. Map Route

Figure III-35

Map



The route map is an ideal choice for visualizing flight routes across Vietnam. It provides a comprehensive view of the aviation network, enabling users to easily identify connections between airports. This tool is particularly useful for evaluating the operational scope of each airline as well as the frequency of specific routes.

The route map displays flight routes through connecting lines between departure and arrival locations on the map of Vietnam. As illustrated in the image, routes originating from Tân Sơn Nhất Airport (Ho Chi Minh City) to other destinations are clearly shown. When users hover over a route, detailed information such as flight code, departure point, and destination is displayed. This provides an intuitive visualization of flight paths.

4.6. Interactivity Usage

Integrating interactivity into the dashboard's visualizations not only enhances user engagement but also enables better exploration, analysis, and decision-making. These interactive features help uncover patterns, trends, and actionable insights, fostering a more effective communication of data for both users and businesses. The dashboard incorporates various interactivity techniques tailored to different types of charts, allowing users to explore data dynamically and intuitively. Below, we detail how interactivity is implemented across the dashboard.

Table III-1

Summarized Insight

CHART NAME	INTERACTIVITY	DESCRIPTION & PURPOSE
Bar Chart	Tooltip	- Tooltips provide dynamic, real-time information when users hover over a specific element (column) in the chart. - In the bar chart displaying "Total Flights by Airline," the tooltip reveals detailed data such as the exact number of flights for the selected airline when the user hovers over the bar.
	Transition	- Transition effects ensure a smooth and visually appealing animation when the data on the chart updates due to filtering or interaction.
	Cross – Chart Interactive	- When a user applies a filter (selecting a specific airline), the bar chart transitions smoothly to reflect the updated data.
	Dynamic Update	- Maintains visual continuity, helping users track changes seamlessly.
	Highlighting	- The bar chart interacts with other visualizations in the dashboard. When selecting an airline update related charts (e.g., total flights or route map).
Scatter Plot	Tooltip	- Provides a cohesive view of the dataset by linking visualizations for better contextual analysis.
	Transition	- The bar chart dynamically updates when users apply filters.
	Cross – Chart Interactive	- Ensures the chart always reflects the filtered dataset.
	Highlighting	- Users can highlight bars representing specific airlines to compare their performance in terms of flight volume. specific airline), the bar chart transitions smoothly to reflect the updated data.
	Dynamic Update	- When hovering over a data point, a tooltip appears showing detailed information about each airline's flight prices.

Heatmap	Tooltip	- Maintains visual continuity, helping users track changes seamlessly.
	Cross – Chart Interactive	- The scatter plot interacts with other visualizations in the dashboard. When selecting an airline update related charts (e.g., total flights or route map).
	Highlighting	- Users can highlight or select specific data points to focus on flights of interest.
	Dynamic Update	- Facilitates comparison between selected airlines or dates, making it easier to analyze pricing patterns for targeted insights.
Map	Tooltip	- Helps identify patterns such as sudden price increases or decreases.
	Transition	- The scatter plot dynamically updates when users apply filters.
	Cross – Chart Interactive	- By hovering over a cell, users can view detailed flight numbers for a specific temporal period.
	Highlighting	- The heatmap interacts with other visualizations in the dashboard. When selecting an airline update related charts.
	Dynamic Update	- When hovering over a cell, the cell is outlined with a bold border to make it stand out from the surrounding cells.

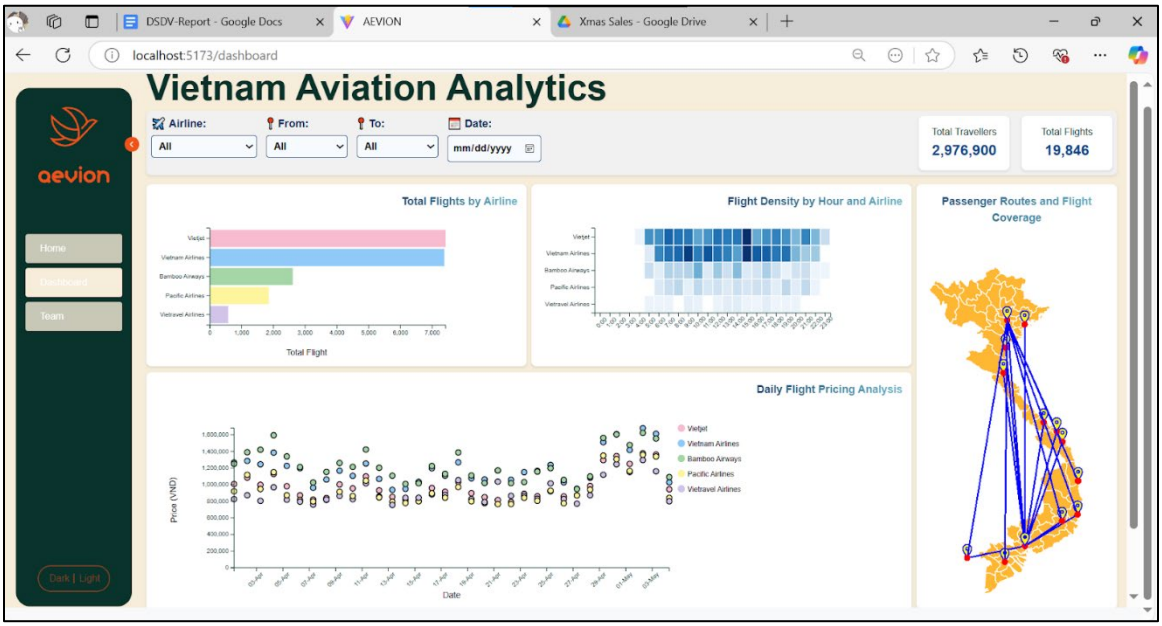
Chapter IV

DEMONSTRATION AND RESULT

The Vietnam Aviation Analytics dashboard's initial interface is thoughtfully designed to offer an intuitive and comprehensive overview of flight data. Its well-structured layout and versatile filtering options empower users to seamlessly explore available destinations, analyze trends, and make accurate predictions.

Figure IV-1

Website



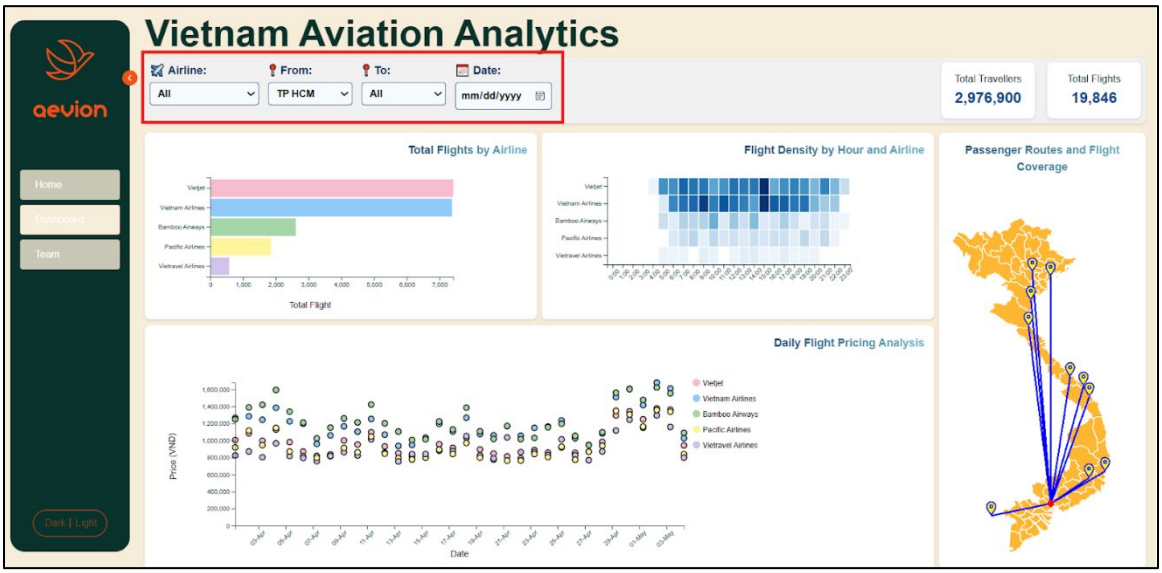
1. Case: From a Single City to All Destinations

When a user based in Ho Chi Minh City intends to utilize the Vietnam Aviation Analytics dashboard to analyze flight data, the filter function becomes an essential tool. Before diving into specific analyses, users may first want to gain a general visual overview

of flight patterns, such as the most active routes, peak travel times, and pricing trends. This initial step allows them to familiarize themselves with the overall aviation landscape. By setting their departure location as Ho Chi Minh City, the dashboard then tailors the information to display relevant routes, destinations, and flight details specific to their point of origin, enabling a more focused.

Figure IV-2

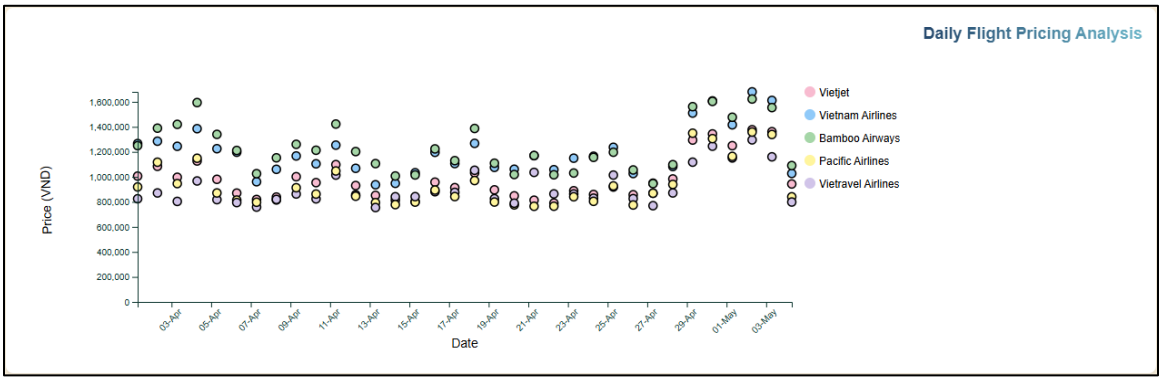
Location Setting



1.1. Daily Flight Pricing Analysis

Figure IV-3

Pricing Scatter Plot



Key Data Visualization:

(i) Price Range: ranges from ~700,000 VND to over 1,600,000 VND.

(ii) Pricing trends vary significantly by airline, with Vietnam Airlines and Bamboo Airways are always the airlines with the highest prices.

Key Insights:

(i) The price increase can be observed closer to the end of April and early May, possibly due to increased demand during holidays or tourist events. It coincides with Vietnam's Reunification Day (30th April) and Labor Day (1st May). This reflects heightened demand during holiday periods.

(ii) Prices are relatively stable and lower during mid-April, presenting a window for travelers to book cost-effective flights.

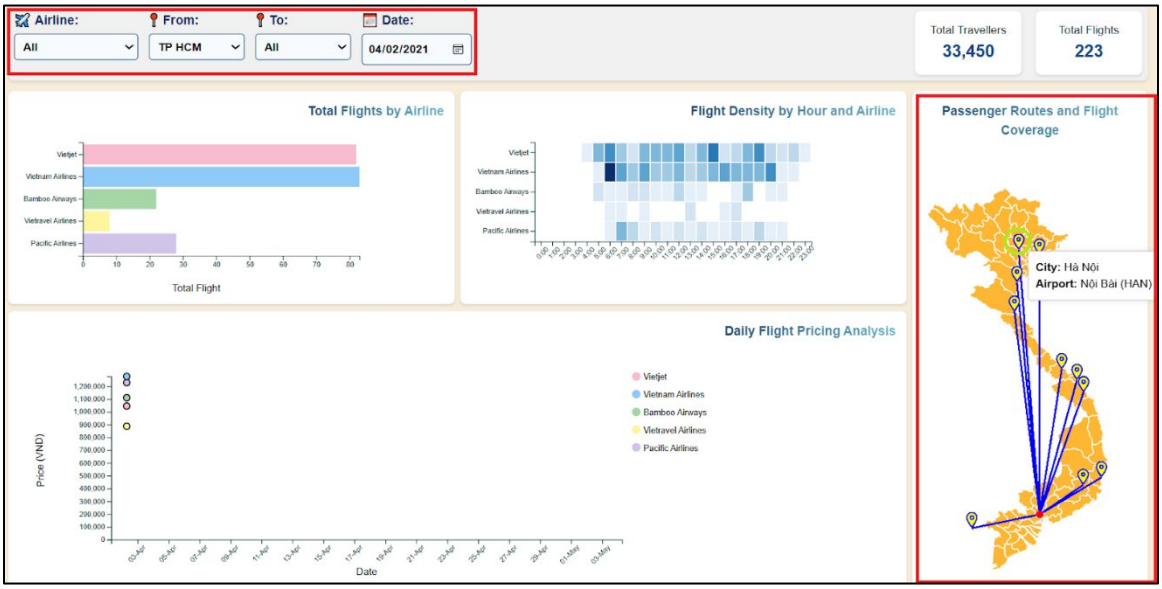
(iii) Vietnam Airlines and Bamboo Airways maintain higher average prices, catering to premium travelers or those prioritizing service quality and flexibility. The other airlines show moderate pricing trends, balancing affordability with service, appealing to travelers looking for a middle ground.

1.2. Filter by Specific Date

Next, the user can refine their search by filtering for a specific date they plan to fly. This helps them check for available flights to their desired destination, such as Hanoi, ensuring they can assess the feasibility of their trip based on the schedule and route availability.

Figure IV-4

Date Setting

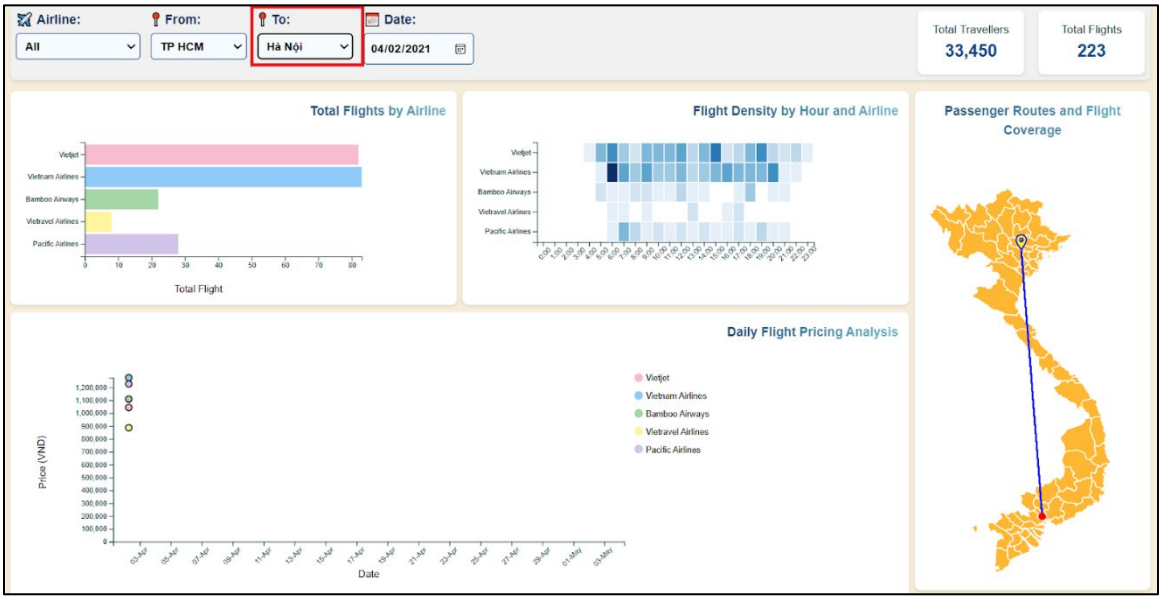


2. Case: From a Single City to a Specific Destination

Having identified on the map that there is a flight route to their desired destination, the user can then proceed to refine their search by selecting the destination on the filter bar. This allows them to extract detailed information. For example, if a user is searching for flights from Ho Chi Minh City to Hanoi, they can select Ho Chi Minh City as the departure point and Hanoi as the destination. This will provide them with specific insights such as flight frequency, ticket prices, and airline options available for this particular route.

Figure IV-5

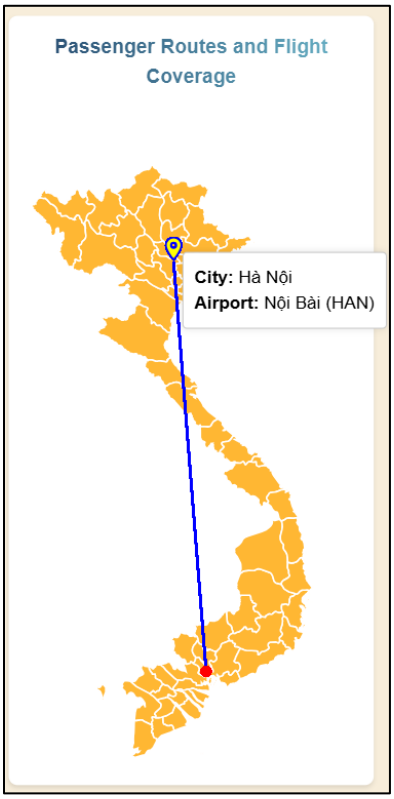
Destination Setting



2.1. Passenger Routes and Flight Coverage

Figure IV-6

Updated Map



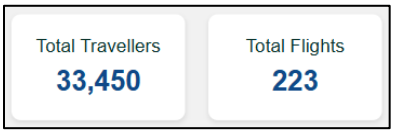
The map clearly illustrates the direct flight route between Ho Chi Minh City (SGN) and Hanoi (HAN).

Relevant information such as city name (Hà Nội) and airport (Nội Bài - HAN) is displayed, confirming the connectivity between these two major cities.

2.2. Total Travelers and Total Flights

Figure IV-7

Counters



Key Insights:

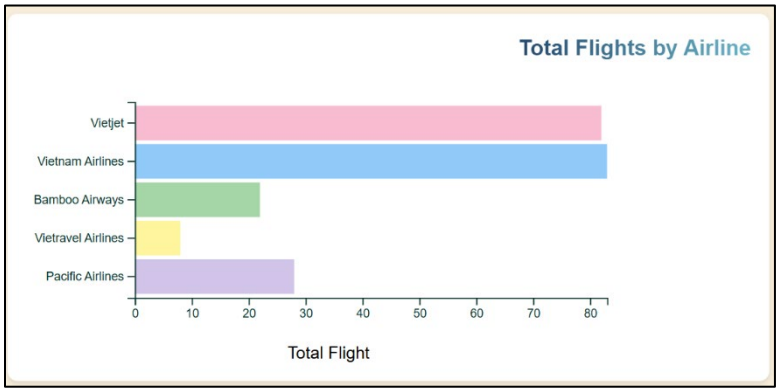
(i) With 33,450 travelers, the data reflects significant demand for flights on this route during this time period. This suggests that customers need to book early to secure seats at reasonable prices as there can be heavy competition for tickets.

(ii) With 223 flights available, there is a significant number of options for travelers to choose from, offering flexibility in scheduling.

2.3. Total Flight by Airline

Figure IV-8

Total Flight by Airline Bar Chart



Key Data Visualizations:

(i) Vietnam Airlines leads with the highest number of flights, closely followed by Vietjet, which ranks second.

(ii) Bamboo Airways, Pacific Airlines, and Vietravel Airlines: Show lower activity compared to the top two airlines.

Key Insights:

(i) The chart highlights Vietjet and Vietnam Airlines as the leading airlines, with Vietjet slightly ahead in total flights. Their dominance offers travelers extensive connectivity and flexible schedules, making them ideal for frequent flyers seeking convenience and access to major routes.

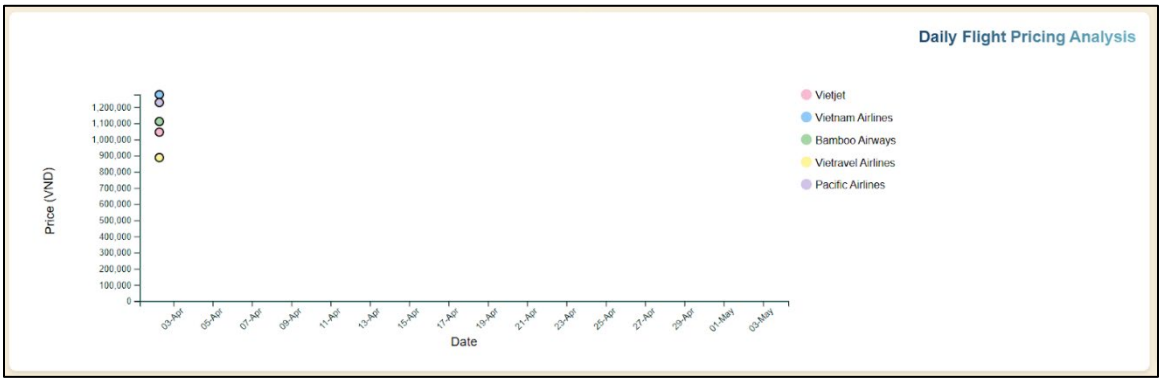
(ii) In contrast, Bamboo Airways, Pacific Airlines, and Vietravel Airlines operate fewer flights, likely focusing on niche markets or less competitive routes. This provides customers with unique travel opportunities or more affordable fares.

(iii) Travelers can choose airlines based on their priorities: Vietjet and Vietnam Airlines for flexibility and frequent schedules, or smaller airlines for cost-effective and niche travel options.

2.4. Daily Flight Pricing Analysis

Figure IV-9

Daily Flight Pricing Scatter Plot



Key Insights:

(i) Vietnam Airlines and Pacific Airlines show the highest ticket prices (~1,220,000 VND), reflecting its premium service. Suitable for travelers prioritizing comfort, reliability, and premium amenities.

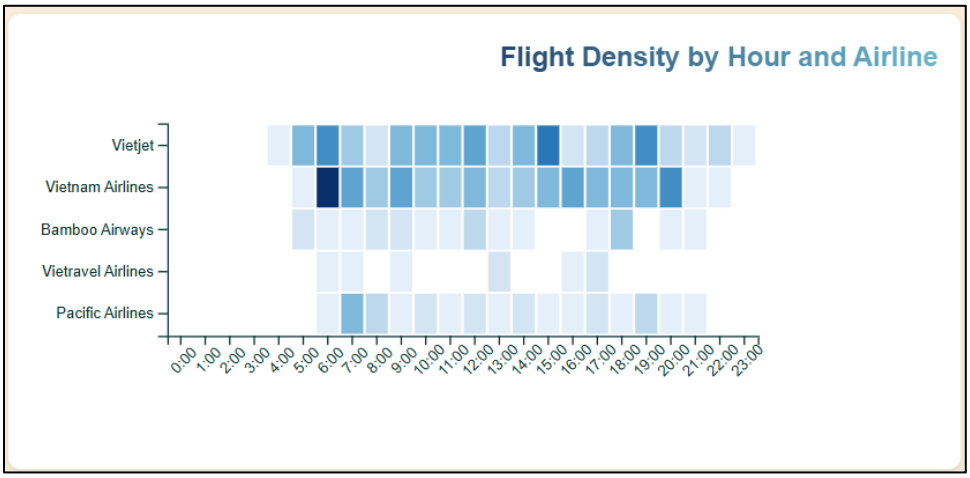
(ii) Vietjet and Bamboo Airways have mid-range pricing (~1,100,000 VND), providing a balance between affordability and availability. These airlines are ideal for budget-conscious travelers who still value flexibility.

(iii) Vietravel Airlines has the lowest ticket prices (~900,000 VND). These airlines are best for travelers focusing solely on cost savings and willing to compromise on flexibility or service options.

2.5. Flight Density by Hour and Airline

Figure IV-10

Flight Density by Hour and Airline Heatmap



Key Insights:

(i) 5:00 AM - 12:00 AM and 14:00 PM - 20:00 PM are the busiest hours for flights, particularly for Vietnam Airlines and Vietjet. These timeframes likely correspond to peak travel demand, such as business trips, work commutes, or leisure travel starting early in the day or after work hours. This suggests that business travelers can choose flights between 5:00 AM - 12:00 AM and 14:00 PM - 20:00 PM to fit in with their regular business schedules and ensure better connectivity.

(ii) Early morning (before 5:00 AM) and late evening (after 20:00 PM) show fewer flights across all airlines.. Flights during these hours may offer a more relaxed travel experience, with less crowded airports and potentially lower ticket

prices. This suggests that leisure travelers can choose flights during off-peak hours for a more relaxed experience and cost savings.

(iii) Vietjet and Vietnam Airlines maintain strong operations during peak hours, while Bamboo Airways, Pacific Airlines, and Vietravel Airlines show reduced activity, likely focusing on niche routes or specific customer needs, appealing to travelers seeking quieter or more affordable options.

2.6. Filter by Specific Date

Figure IV-11

Filtered Display

The dashboard allows users to filter by individual airlines, making it easier to select a suitable carrier and view detailed information on pricing and peak hours' fluctuations across specific days and times, as illustrated above. This feature provides an overview of pricing trends and flight density unique to each airline.

3. Summary

To make an effective decision, travelers should align their priorities, such as cost, flexibility, comfort, or schedule, with the options provided by various airlines and time slots. For those with tight schedules or business needs, Vietjet and Vietnam Airlines offer the best choices during peak hours due to their high flight frequency and flexibility.

Budget-conscious travelers can benefit from Vietjet or Bamboo Airways, which provide mid-range pricing and a balance between affordability and availability. Alternatively, Vietravel Airlines, with the lowest fares, is ideal for those willing to compromise on schedule flexibility to save on costs.

For passengers seeking a premium experience, Vietnam Airlines and Pacific Airlines stand out with their enhanced comfort, reliability, and premium services, despite their higher fares. These airlines cater well to those who prioritize superior service and amenities during their journey.

Lastly, leisure travelers or those with flexible schedules should consider flying during off-peak hours with Bamboo Airways, Pacific Airlines, or Vietravel Airlines. This ensures a quieter, more relaxed journey with potential cost savings.

Chapter V

CONCLUSION

1. Achieved Goals

The project has successfully achieved the goal of developing an intuitive web interface with a clear and easy-to-navigate layout. The design, with user experience in mind, ensures that end-users can easily access desired information without feeling overwhelmed. By prioritizing simplicity and functionality, this layout allows users to focus on their primary intention for visiting the website.

The dashboard, the central developing component, serves as a reliable tool to help users make informed and cost-effective flight booking decisions.It provides detailed information on ticket prices, flight schedules, route coverage, and peak travel hours. With this data, users can easily compare ticket prices across airlines, identify peak travel hours, and track pricing trends through interactive charts and flexible filters. This central functionality empowers users to make practical and well-informed flight choices based on real-time data, selecting optimal prices and schedules.

In addition to achieving the project's functional goals, the development process also provided an opportunity to practice web design technologies. The project has utilized HTML, CSS, Javascript, D3 library, Vite with React, and Tailwind CSS to create a responsive and dynamic user experience. The integration of these tools has supported seamless development and provided flexibility in adapting to project requirements.

Furthermore, the project serves as a valuable opportunity for teamwork. All team members collaborated effectively, coordinating their efforts, communicating openly, and contributing to various aspects of the project, all of which ultimately led to the successful completion of the website. This collaboration has enhanced both the technical and interpersonal skills of all involved.

2. Future Work

While the website meets its current objectives, there are several areas where future improvements and enhancements could be made to further increase its functionality and user experience. One important addition would be the integration of real-time flight data, allowing users to access up-to-date information on flight availability, ticket prices, and schedule changes. This would significantly improve accuracy and relevance for users so that they can make decisions based on the most current data available.

Another potential improvement involves utilizing advanced technologies to handle the data structure more efficiently, particularly as the dataset grows too large to load effectively on the web. By employing solutions such as server-side data processing, cloud databases, or distributed computing frameworks, the website could handle large volumes of data seamlessly without compromising performance. This would complement the integration of real-time data and ensure smooth user interactions.

Additionally, the functionality to generate insightful reports based on selected comparisons. Users could benefit from having the option to download or view detailed

reports of their comparisons. This would provide more in-depth analysis and a more comprehensive view of the data, aiding users to make better decisions.

Refining the web layout and improving the overall design are other avenues for improvement. A more streamlined and intuitive interface, coupled with enhanced navigation, could make the website more user-friendly and accessible. Optimizing the responsiveness of the layout for different devices would ensure a smooth experience for users across desktops, tablets, and mobile phones.

By incorporating these changes, the website ultimately would benefit from increased web traffic, engaging more users, and a better value proposition while providing the best possible, user-friendly experience.

LIST OF REFERENCES

[1] GeeksforGeeks. (n.d.). *How to add effects to tooltips?*. Retrieved from https://www.geeksforgeeks.org/how-to-add-effects-to-tooltips/

[2] W3Schools. (n.d.). *Style transition property*. Retrieved from https://www.w3schools.com/jsref/prop_style_transition.asp

[3] Rahman, M. (2021). *Optimizing tooltip design for data visualization*. Retrieved from http://essay.utwente.nl/85856/1/rahman.pdf

[4] IOP Science. (2021). *Tooltip effects in user interaction studies*. *Earth and Environmental Science, 1151*(1), 012049. Retrieved from https://iopscience.iop.org/article/10.1088/1755-1315/1151/1/012049/pdf