­­­



*Capable People, Capable Communities*

**Micro-Credentials in Software Development**

**Certificate in Software Development**

**MCSD51**

**Project Report**

**Airscammer**

**Scam Reduction Education App**

**Felicia**

**September 2022**

Contents

[**Executive Summary** 3](#_Toc114130649)

[**Introduction** 3](#_Toc114130650)

[**Initial Proposal** 3](#_Toc114130651)

[**Variations In User Requirements From The Initial Proposal** 3](#_Toc114130652)

[**Design Documents** 11](#_Toc114130653)

[**Developer Documentation** 15](#_Toc114130654)

[**Tools and Technologies Used (in alphabetical order)** 16](#_Toc114130655)

[**Overview of software and its source code** 16](#_Toc114130656)

[**Known bugs and limitation** 18](#_Toc114130657)

[**SDLC** 18](#_Toc114130658)

[**Project timeline** 18](#_Toc114130659)

[**Ethical and cultural impact** 19](#_Toc114130660)

[**Test Document** 19](#_Toc114130661)

[**User Documentation** 20](#_Toc114130662)

[**Configuration Guide** 21](#_Toc114130663)

[**Future Software Enhancement List** 21](#_Toc114130664)

[**Reflection** 21](#_Toc114130665)

[**Appendix: Source Code** 22](#_Toc114130666)

# Executive Summary

The most common air travel scams are the ones that occur in an online space. It is easy for scammers to build a fake airline website with very similar text and images to persuade users and take away their money.

Airscammer is a web application that simulates what a potentially suspicious booking website will typically do, the name Airscammer is inspired by "Skyscanner" an actual aggregator airline website. The purpose of this project is to educate my user to pay attention to online booking by setting up some "traps" for them and informing them what potential harm could come from those "traps".

This web application is developed with HTML, CSS, Bootstrap, MYSQL, Node.JS, Passport and JavaScript. This is a self-sponsored, individual project developed in a span of 3 weeks with iterations each week. During the 3 weeks period I was able to push myself and add some additional features such as showing local time when user is logged in and deploying the application to a cloud server.

# Introduction

# With the coronavirus vaccines rollout and the decrease in travel restrictions, people are looking to travel again. The demand on finding cheap flight tickets increases and it's an opportunity for scammers to exploit.

# Usually, the scam will start with a person searching for cheap flights and then coming across a great deal at an unbelievable price. After booking the flight, the customer notices they never actually received a ticket.

# It is quite common for people to get scammed by these websites, I've seen some of my family members and friends fall into these scams. I am aware that there are a lot of online resources about airline online scams, but not many people are very keen on reading articles. With Airscammer I aim to make learning more fun, instead of reading a bunch of articles and not fully understanding them, we can just learn by doing.

# Initial Proposal

# Scope

# The completed project will have at least four web pages: a home page, a page on choosing the flights, a page to collect customer information and a page to show user's choice statistics. The completed project will also have a database to compile the information on the user's choice of flight, dates, destination, and customer information.

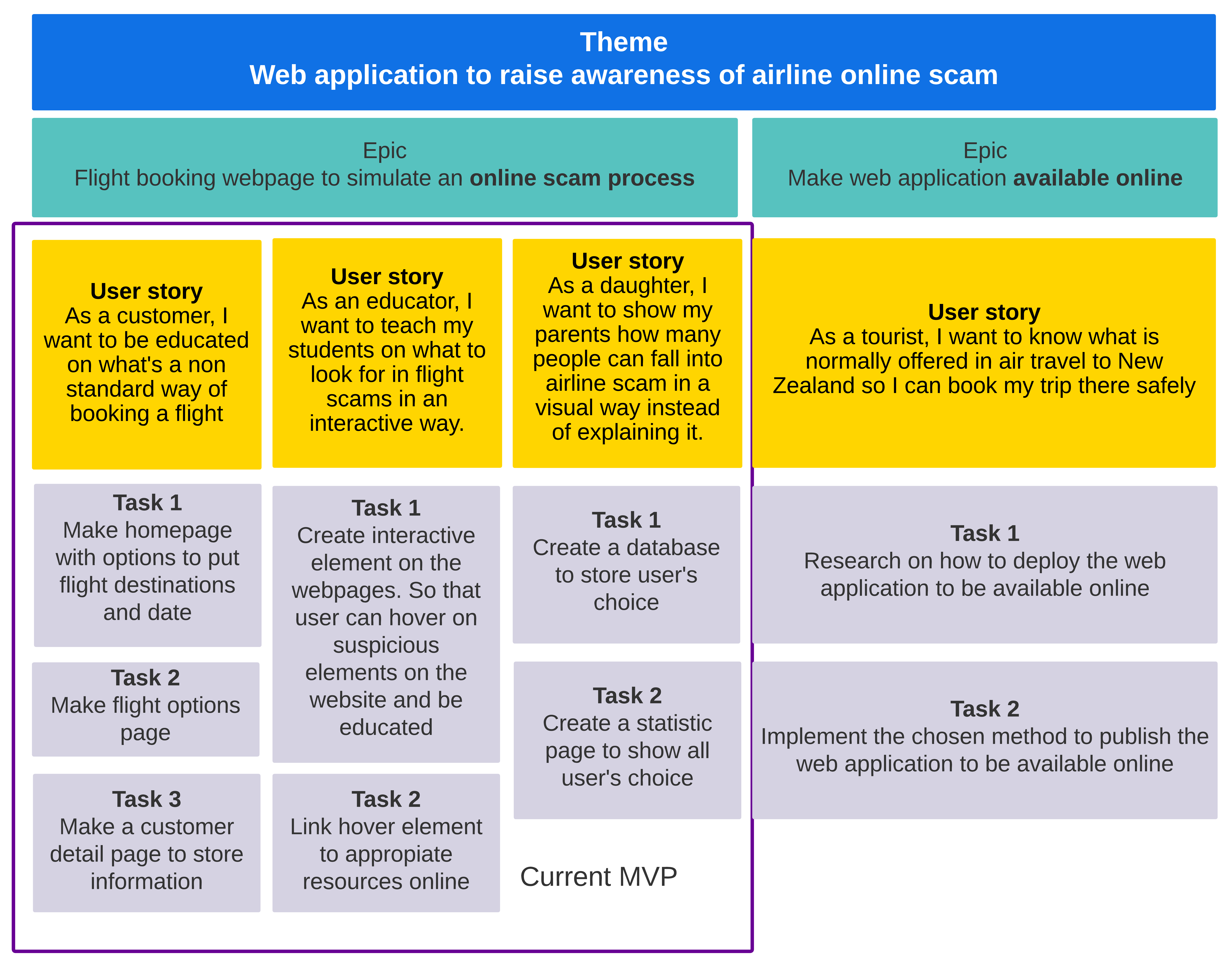
# User stories and use cases

User stories:

* As a customer, I want to be educated on what's a non-standard way of booking a flight.
* As an educator, I want to teach my students on what to look for in flight scams in an interactive way.
* As a daughter, I want to show my parents how many people can fall into airline scam in a visual way instead of explaining it.

As a tourist, I want to know what is normally offered in air travel to New Zealand so I can book my trip there safely.

User story map:



# Ethical and cultural impact

# I know that different groups in society experience benefits and harm in different ways. This project strives to promote considerations of equity as a fundamental element of impact assessment. This project seeks to take information to measure the typical behaviour of people doing online airline booking, and we are especially mindful of the concerns of privacy, and we do suggest for our users to not fill in their real details for demonstration purposes. All the information shared in the forms will be stored securely and not leaked out to the public.

# SDLC

# In this project, I will be using the Agile method, Agile lets me develop projects in small increments and this methodology is good for teams that need to finish specific projects quickly. Features and functionality can be added or modified with relative ease to accommodate the often-inevitable changes in user requirements and preferences.

# Project timeline

I completed the MVP in sprint 1. I focused on creating all the HTML files for the pages with no CSS. Then create the database on the MySQL server, I assigned more time in this area since I'm not quite familiar with the database system. After the database is created then I created a page to show the user's data statistics. The next sprint will cover more on the interactive part of the web application.

Sprint 2 I focused on making the web application more interactive, I added more features such as a CSS tooltip when the user hovers over a certain "suspicious element" and the tooltip will redirect the user to appropriate resources on online scams when clicked.

Sprint 3 (week 3) everything works according to the plan on sprints 1 & 2, on sprint 3 I did more research on how to make this web application available online, so it is accessible for the public to use.

Graphical user interface, timeline

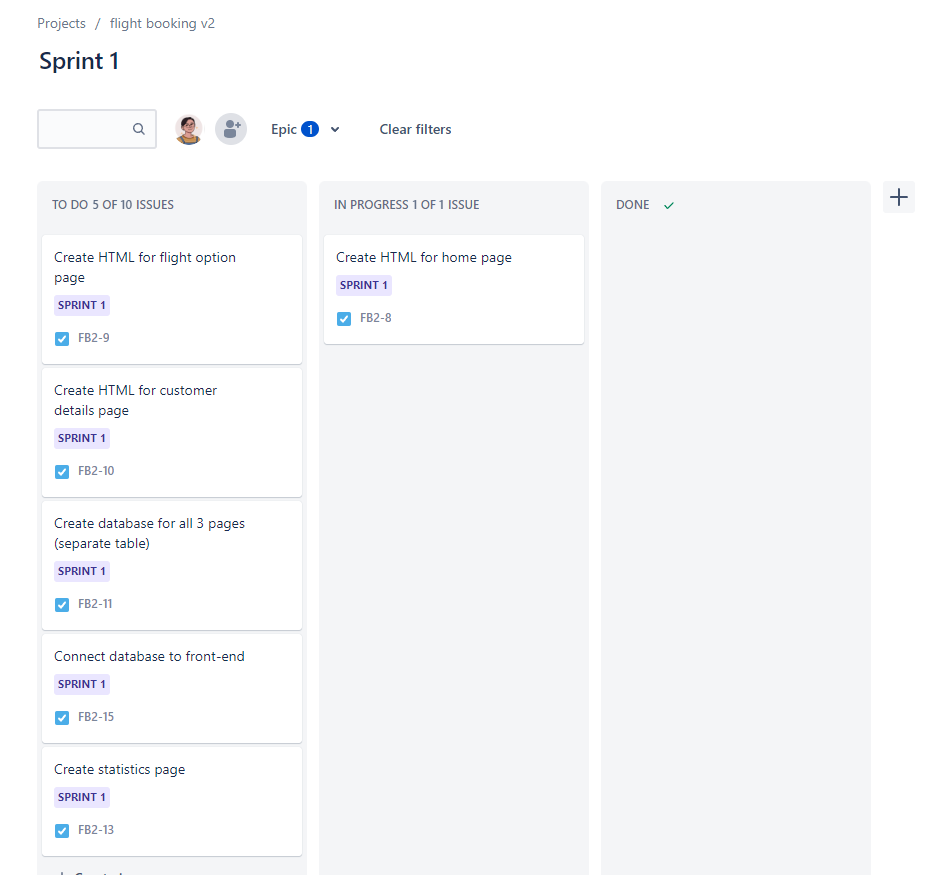
Description automatically generated with medium confidence

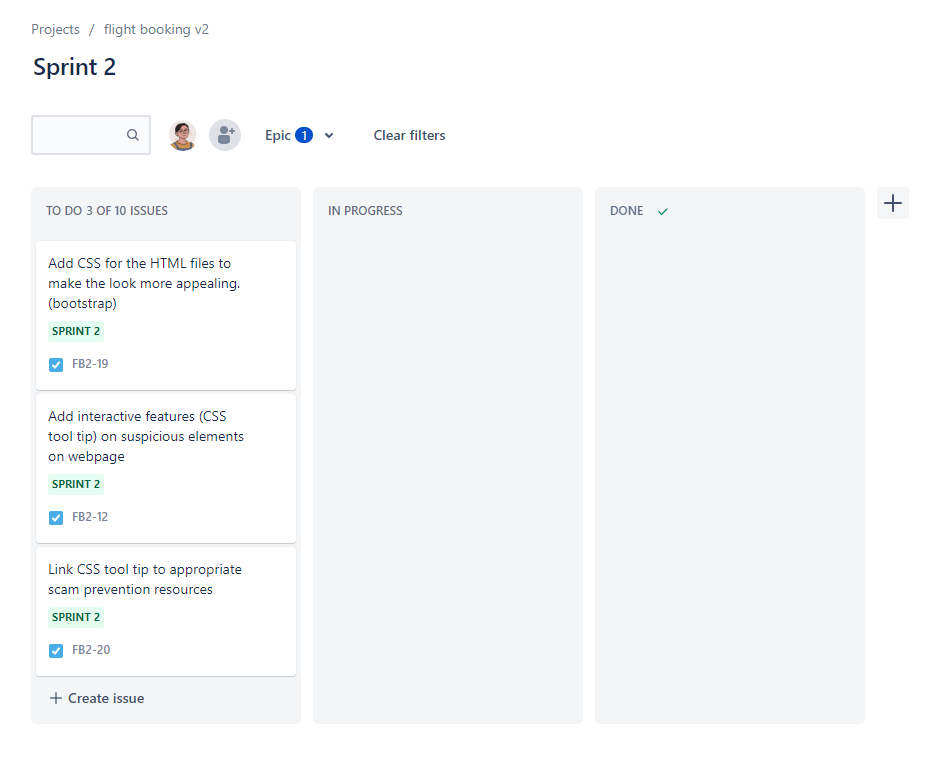
Sprint 1 : Aug 31, 2022 - Sept 09, 2022

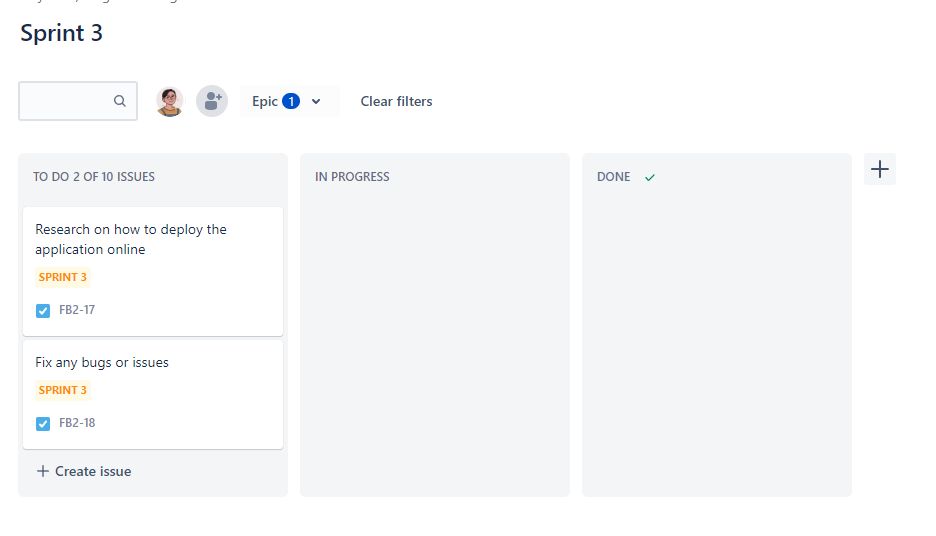
Sprint 2 : Sep 10, 2022 - Sep 18, 2022

Sprint 3 : Sep 19, 2022 - Sep 28, 2022

Link : [**https://applescan.atlassian.net/jira/software/projects/FB2/boards/2/roadmap**](https://applescan.atlassian.net/jira/software/projects/FB2/boards/2/roadmap)

****

****

****

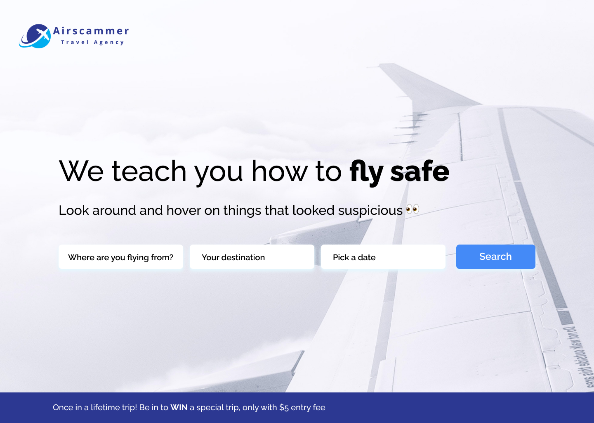
# Diagrams

Diagram

Description automatically generated­­

The booking process is as follows:

1. Users go to the homepage, pick the date and destination, and click search flight.
2. Users will be taken to choose flights (this is the interactive part when users will be pointed out things that looked suspicious).
3. After the user chooses the flight, they will be taken to fill in the customer detail (customer details will ask for suspicious questions and the user will be made obvious to educate them on this topic).
4. Users click “submit”.
5. Save each response into the database.
6. Back-end compute and display statistics on the results on the screen to the user.

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generatedGraphical user interface, application, Teams

Description automatically generated

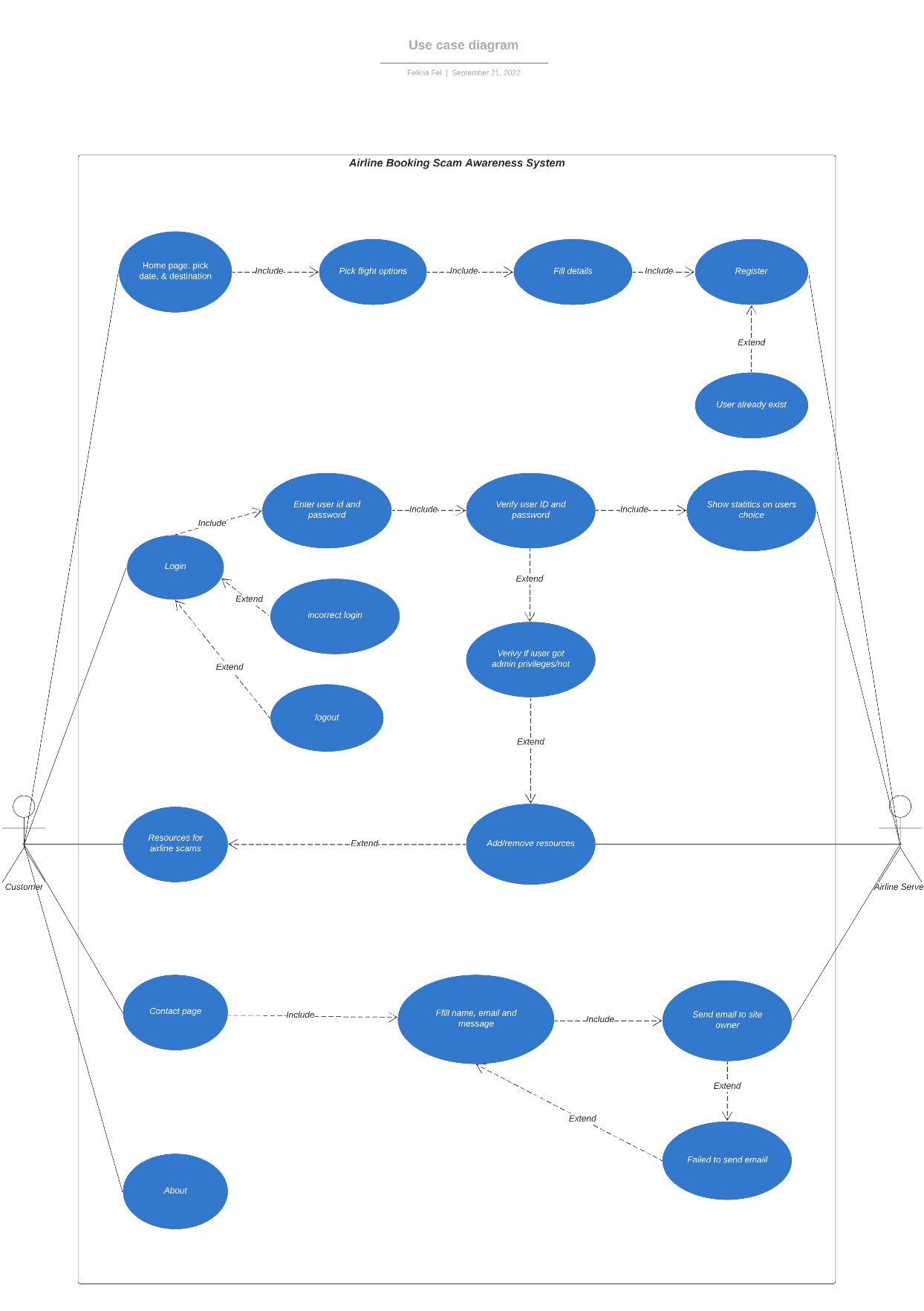
Working prototype: <https://www.figma.com/proto/ou5keLEY9P0isEsqn60PjI/AirScammer?node-id=1%3A2&scaling=min-zoom&page-id=0%3A1&starting-point-node-id=1%3A2>

# Variations In User Requirements From The Initial Proposal

During the development process, I made some changes to the user requirements and application features. After some discussion with my lecturer and some potential users, we decided to implement a secure login system that can take care of the security of my user's account and information.

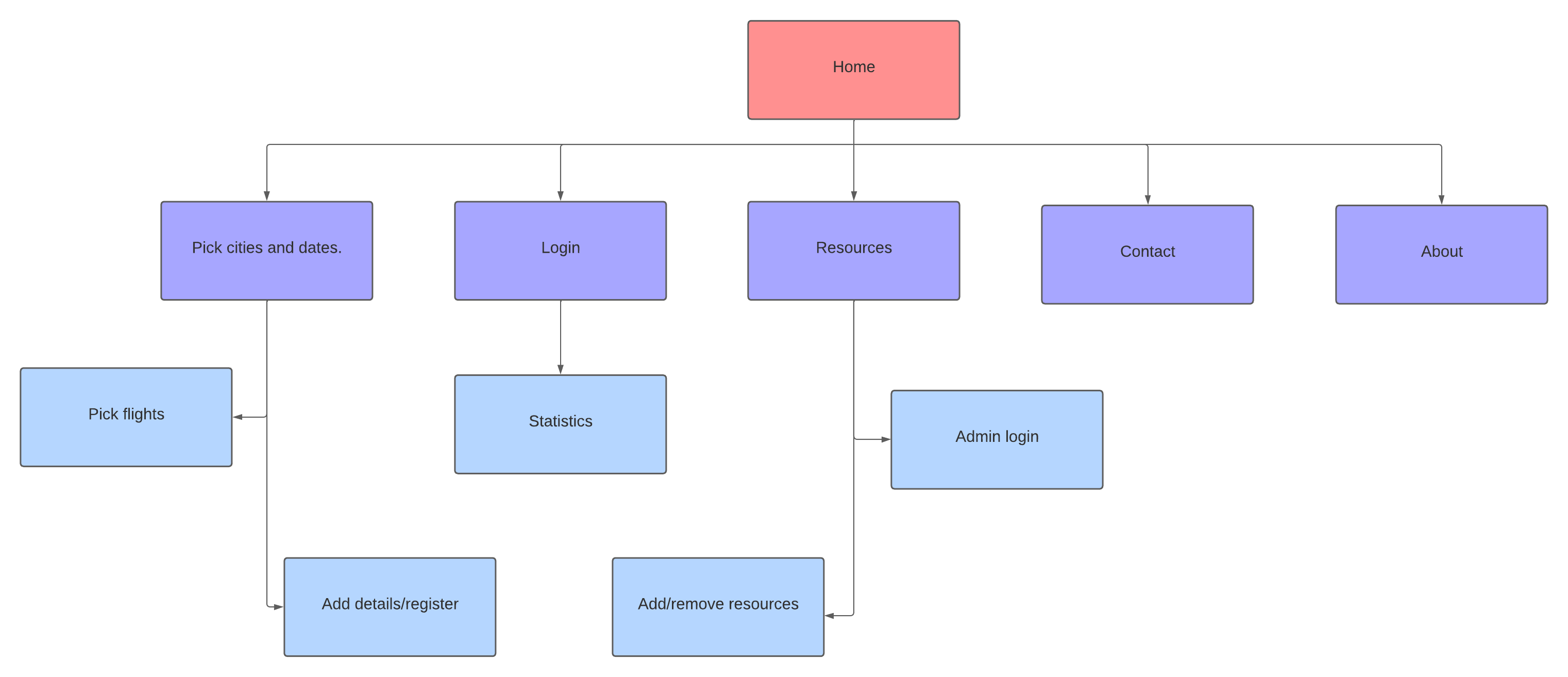
Collaboration is an important part of this project, and to gather feedback from my users I also added a contact system, so my user can directly send me messages & feedback from the web application. I also have added a resources section that can be accessed by the web administrator so the resources on online scams can always be updated.

New use case diagram:

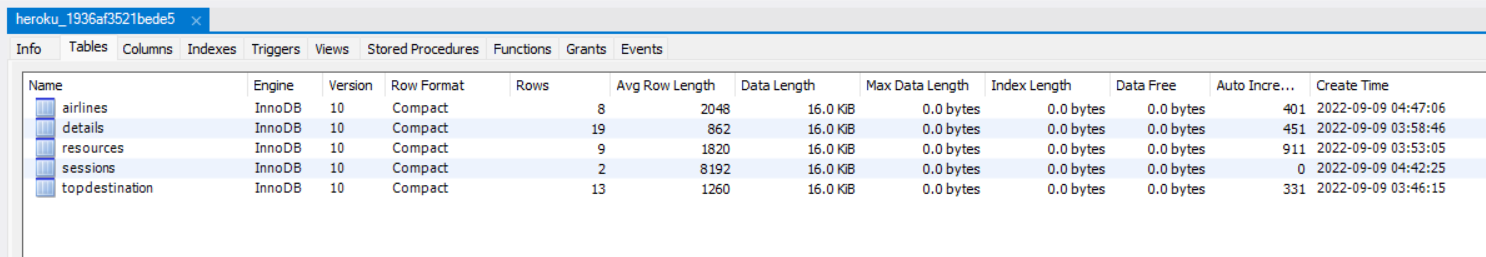


# Design Documents

Sitemap



Database structure.



Tables:

1. **airlines**: store user’s selection of airlines from <https://airscammer.herokuapp.com/flight>
2. **details**: store user’s details from the registration page <https://airscammer.herokuapp.com/details>
3. **resources**: containing all the resources from <https://airscammer.herokuapp.com/resources>
4. **sessions**: automatically stores user’s session with randomized cookie session key.
5. **topdestination**: store user’s choice from <https://airscammer.herokuapp.com/>

Graphical user interface

Description automatically generated

Columns

1. **airlines**:
   * id: id number
   * airline: containing airline names from user selection
2. **details**:
   * id: id number
   * username: username from the registration form
   * hash: automatically encrypted from passport
   * salt: automatically encrypted from passport
   * dogName: dog name from the registration form
   * momName: mom name from the registration form
   * isAdmin: if the value is 1 then user is admin but if it’s 0 then they are not admin
3. **resources**:
   * id: id number
   * source: article source
   * url: article url
4. **sessions**:
   * session\_id: randomly generated ids
   * expires: containing time when the user will automatically get logout
   * data: user’s credentials (to determine if the user logged in or not)
5. **topdestination:**
   * id: id number
   * cityTo: user’s city destination
   * cityFrom: user’s city from

Graphical user interface, text, application

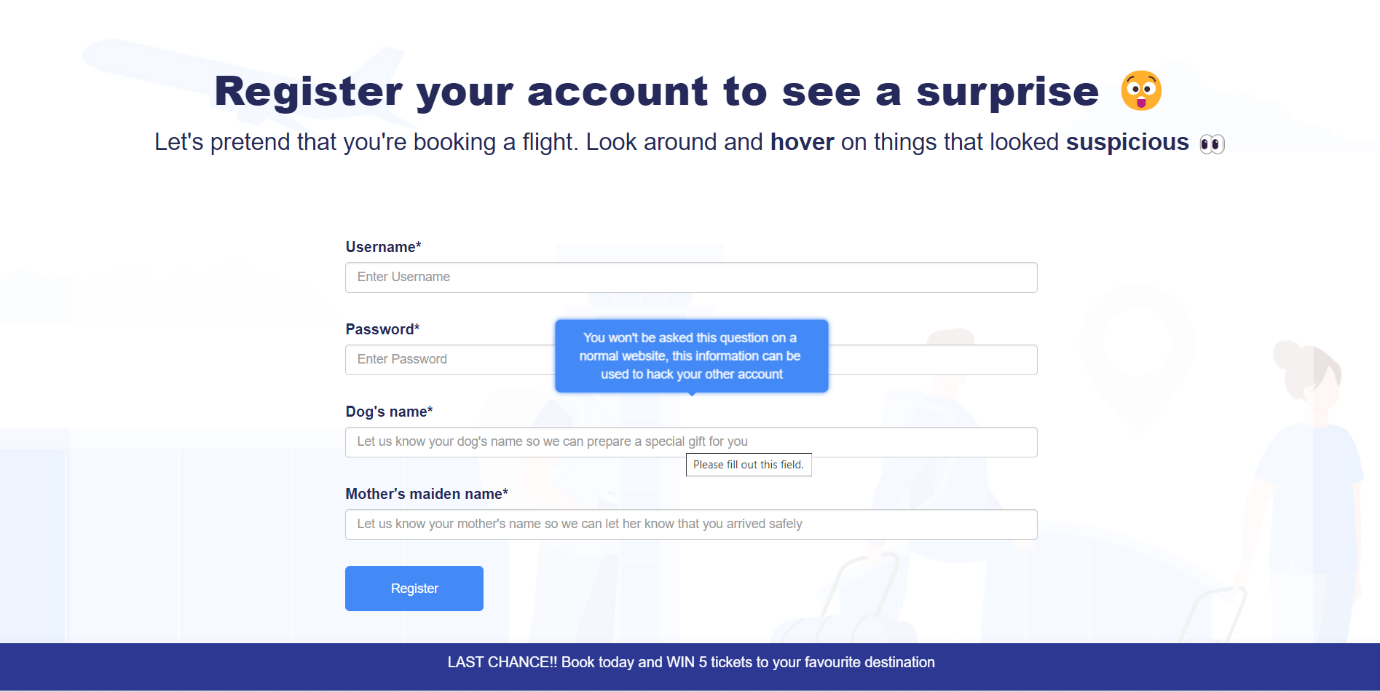
Description automatically generated

Home page – When the user hovers over various elements, a tooltip will pop out and tell users to pay attention to these items.

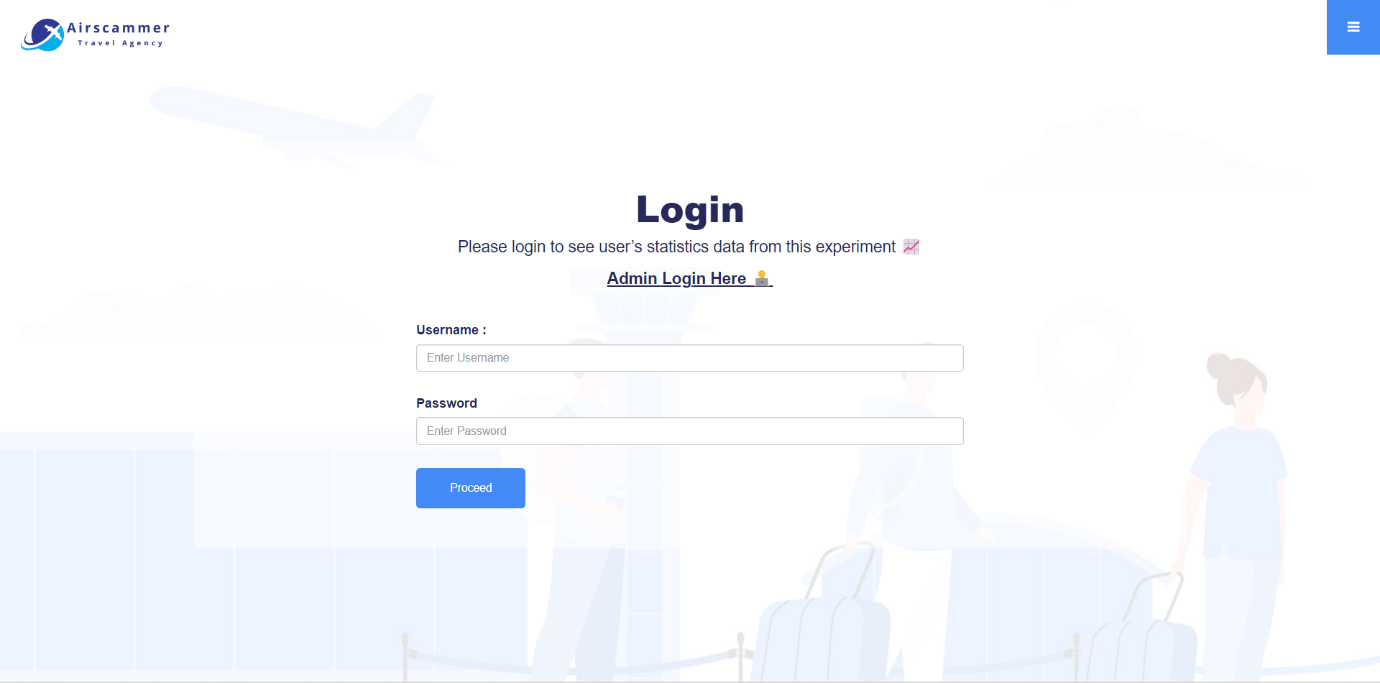
Graphical user interface, application

Description automatically generated

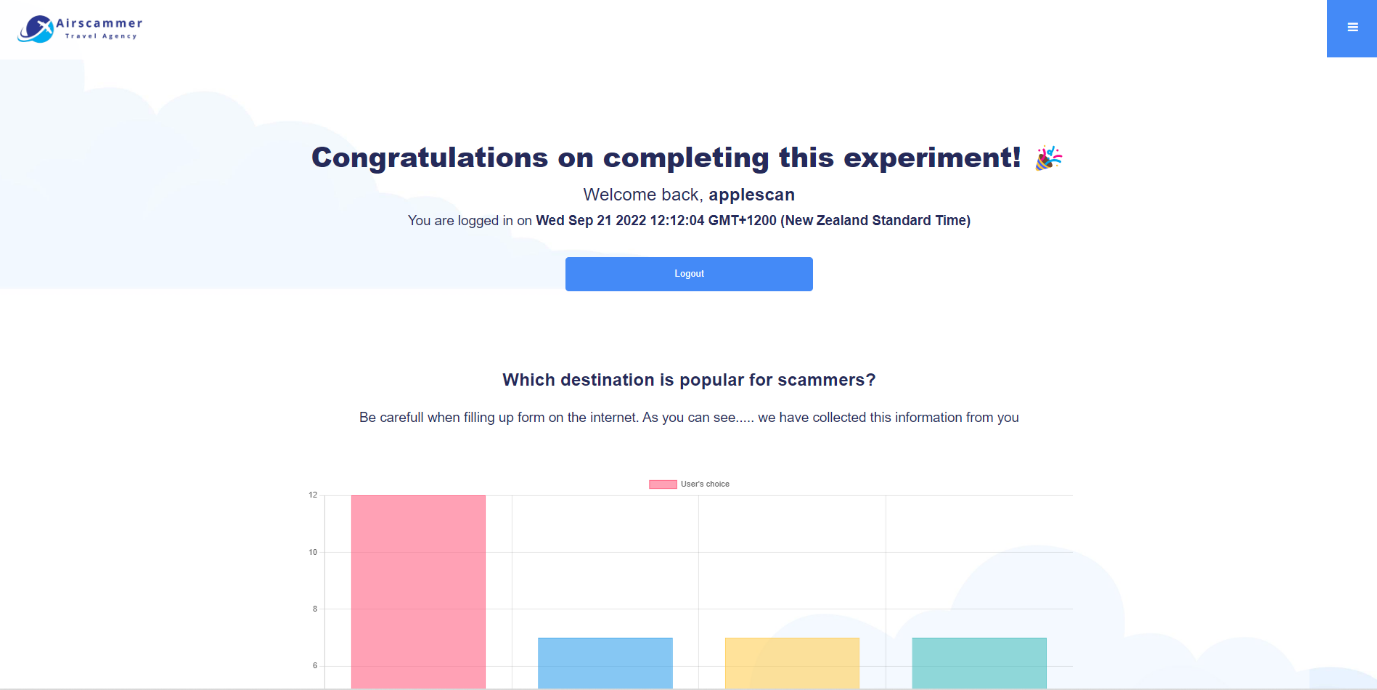
Flight Option page – When the user hovers over various elements, a tooltip will pop out and tell users to pay attention to these items.



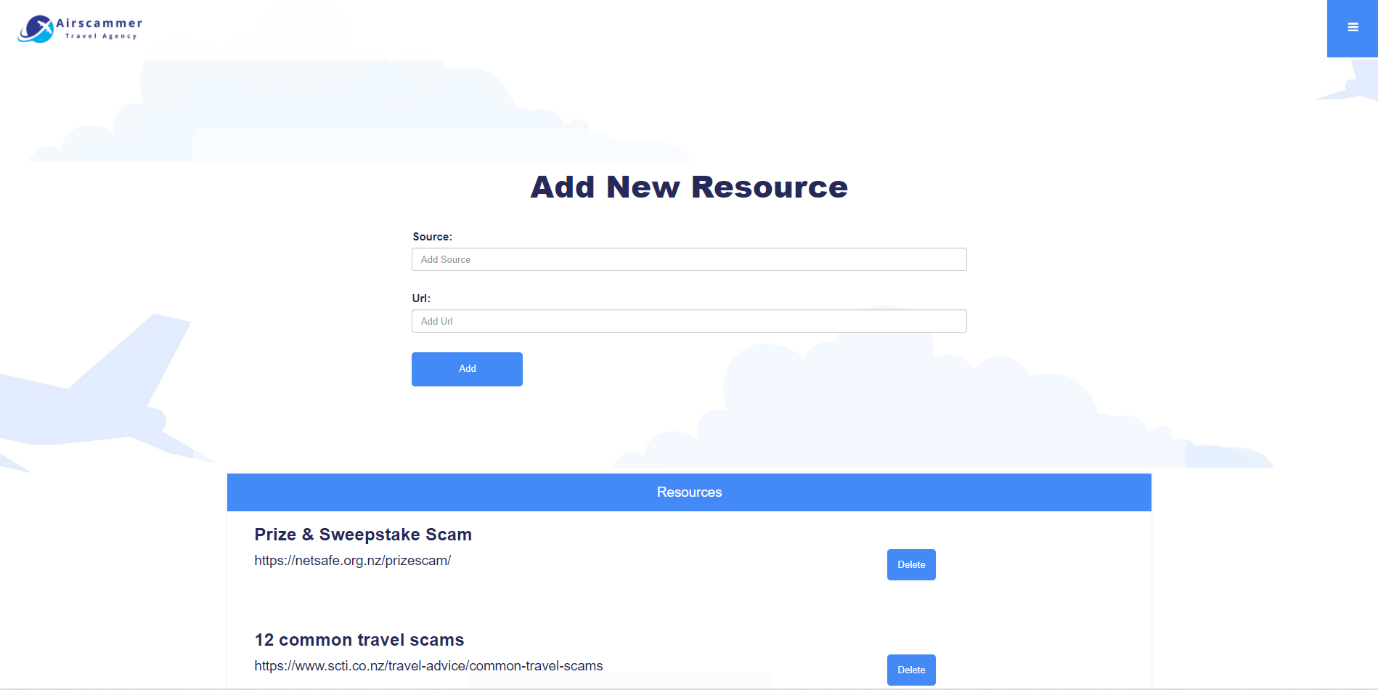
Detail registration page – When users hover on various elements, a tooltip will pop out and tell users to pay attention to these items. User will create their account here.



Login page – The user will log in to see the other user's choices and some statistics on how they compare to the others and hopefully learn from those data.



Statistics page – Username and login time and location will show on the page.



Add new resource page – Only the administrator can log in to this page and add/remove resources.

Icon

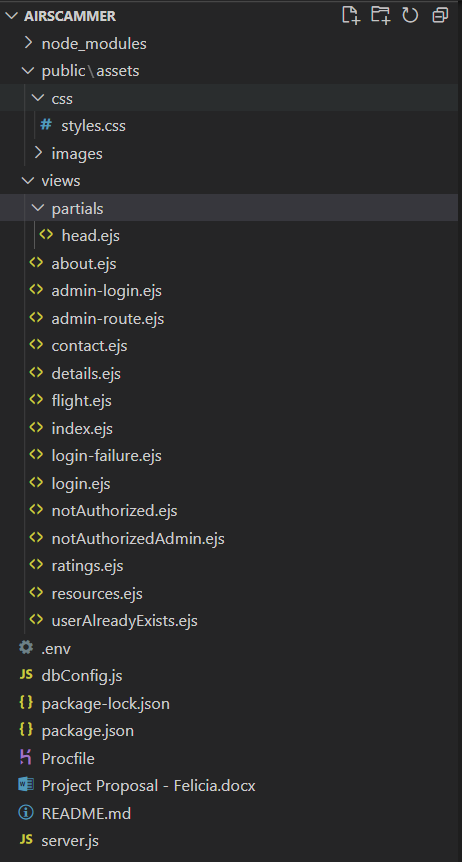
Description automatically generated with low confidenceUser details database – if the column "isAdmin" = 1 then that user is an administrator. All password is fully encrypted with hash and salt.

# Developer Documentation

# Tools and Technologies Used (in alphabetical order)

* Bootstrap
* CSS
* Clear DB
* Chart.js
* Cookie-parser
* EJS
* Express.js
* Figma
* GitHub
* Heroku
* HTML
* JavaScript
* Lucid Chart
* My SQL Workbench
* Node.js
* Nodemailer
* phpMyAdmin
* SendGrid API
* XAMPP
* Visual Studio Code

# Overview of software and its source code



From top to bottom:

1. **Public**: contains public assets such as CSS and images that is used in the web application.
2. **Views**:
   1. **Partials**: contains the universal head element of all the ejs files in the views folder.
   2. **About.ejs**: code for the about page (static webpage).
   3. **Admin-login.ejs**: Login page for admin to see admin-route.ejs
   4. **Admin-route.ejs**: a page that can only be accessed by admin. Admin can add/remove resources from this page.
   5. **Contact.ejs**: user can send email to my email address by filling up the contact form and click send. I use nodemailer and the SendGrid API for the contact form.
   6. **Details.ejs**: User fill up their information and register so they can login and see the ratings.ejs.
   7. **Flight.ejs:** User pick a flight and the data will be recorded in the database and shown in ratings.ejs.
   8. **Index.ejs:** Homepage for the web application. User can pick flight destinations and date, the data will be recorded in the database and shown in ratings.ejs.
   9. **Login-failure.ejs**: This page will show up if user input the wrong password/username.
   10. **Login.ejs:** Login page to see ratings.ejs.
   11. **NotAuthorized.ejs**: This page will show when user is not logged and try to see the ratings.ejs / admin-route.ejs.
   12. **Ratings.ejs**: This page takes all data from the database. (Inputs from index, flights, and details) and show them in charts and tables. Only logged in user can see this page.
   13. **Resources.ejs**: A page showing all resources about airline scam. Resources data is gathered from database. Admin can add/remove resources from the admin-route.ejs without having to directly go to database to add/remove resources.
   14. **userAlreadyExist.ejs**: This page will show if user fill up an already existed username in details.ejs page.
3. **.env**: a file containing my SendGrid API code.
4. **dbConfig.js**: File to configure database connection.
5. **Package-lock.json**: package file
6. **Package.json**: file containing all settings for the node application, including dependencies.
7. **Procfile**: File specifically used when deploying an application to Heroku. Contains settings on what to run.
8. **Project Proposal:** Original project proposal.
9. **Readme.md:** File containing a little information, used for GitHub repository.
10. **Server.js:** File containing the server to run the node application.

**Server-side:**

I'm using a free database add-on from Heroku called Clear DB, the database can be accessed using MYSQL Workbench or any other application using these credentials:

**username**: (not available in this sample document)

**password**: (not available in this sample document)

**host**: (not available in this sample document)

**database**: (not available in this sample document)

# Known bugs and limitations

# **Problem**: Issues with using emojis as usernames, there are no specific rules on what is allowed to be used as a username from the input field. The database would only record varchar inputs, so the emojis are not recorded in the database.

# **Cause**: When user put their emoji username and password on the login page, it will always return as a false since there's no record of the emoji username in the database.

# **Solution**: I will fix this by adding some rules to the username field to prevent users from creating invalid usernames.

# **Limitations:** The web application lacks the forgot password" feature. Due to time constraints, I am unable to create this feature yet. I'm aiming to create this feature soon, as it is important for the users to be able to change their password when they forgot their original one.

# SDLC

The agile methodology is very effective for this project, Agile lets me develop projects in small increments and lets me focus on the MVP first before going through with other parts of the application. Features and functionality can be added or modified with relative ease, the 1-week sprints are very effective for me as it motivates me to finish quickly and reach the next goal. I'm able to complete the project earlier than expected and additional features to my web application.

Project timeline

A picture containing graphical user interface

Description automatically generated

# I managed to complete the project a bit earlier than the expected timeframe. I had planned sprint one to be all about finishing major issues with the program and completing all the technical debt on sprint 2 while designing the UI. Each sprint has been executed greatly with no major problems, I was able to get help and feedback from my tutor, classmate, and peer tutor on Saturdays.

# During the extra time that I have, I'm doing user testing for various people that I know by sending them the link to my web application, with their feedback I will try to fix minor issues that can be fixed during the last week before the final presentation date.

# Ethical and cultural impact

# During the development process, I have learned more about the impact of online scams on the public, and I understand greatly that not everyone can distinguish the difference between a real website and a fake one, so I created the resources page and make sure that it’s visible on the menu, so users can read for more information. All the items in this web application are made to be as fake as possible so there’s no room for misinterpretation.

# As security is a very important factor, I make sure that all the data saved in the database is not shared with the public and users can ask for their information to be removed by contacting me from the contact form.

# Test Document

I had several of my friends and family from New Zealand and Indonesia try out my web application, my testers are free to do anything, and I challenged them to try to break the application, here are the results of user testing:

Tester 1: The user added the year 20900 in the date selection, which is allowed in the web application. But it is uncommon to book so far ahead, the solution is to add the max date rule in the calendar.

Tester 2: The user tried to pick the same destination going from and going to, this is not a big issue, but can cause a misunderstanding. The best thing to do is to not allow this to happen by setting a rule on city selection.

Tester 3: The user tried to use emojis as a username, but the database would only record varchar inputs, so the emojis are not recorded in the database. When user put their emoji username and password on the login page, it will always return as a false.

Tester 4: The user added empty characters as a username and the user can create an account, this should not happen.

All these issues are not fundamental to the workings of web applications, and they can be fixed fairly quickly to improve the user's experience.

Graphical user interface

Description automatically generated with medium confidence

# User Documentation

Visit [**https://airscammer.herokuapp.com/**](https://airscammer.herokuapp.com/)to access the web application.

**Booking**:

1. Go to the homepage, pick the date and destination, and click search flight.
2. Users will be taken to choose flights.
3. After the user chooses the flight, they will be taken to fill in their detail (customer details will ask for suspicious questions and the user will be made obvious to educate them on this topic).
4. Click “Register”.
5. The user will be taken to the login page to see the statistics data.
6. Login to see the statistics data.
7. In case you forgotten your password please go to the **contact page** and let me know.

**Contact page:**

1. Fill in the details and message
2. Click “send”
3. Message will be sent to my email address.

# Configuration Guide

**Updating the resource page process (admin only):**

1. Go to resources page
2. Click add resources
3. Login as an administrator
4. After login admin can add new resources or remove or existing one

# Future Software Enhancement List

# The next version of Airscammer will have a "forgot password" feature so users can automatically reset their password instead of emailing the admin to change their password.

# Reflection

I started with a little knowledge of HTML, CSS, and some experience in building static websites in WordPress and Wix.

By doing this project I learned how to expand my basic knowledge of HTML & CSS into a fully working web application. I expanded my knowledge of SDLC, server-side and client-side web applications and learned how to use Bootstrap, MYSQL, Node.JS, Passport, user authentication and JavaScript. I am very happy to be able to learn about creating a web application with a database and deploying it online, it is the next step in my learning journey to fully expand that knowledge and build more applications with back-end.

I am aware that there is much more to learn, and I am very motivated to learn more about APIs, React, and other frameworks in the future.

With the agile methodology I learned from this course, I improved my project and time management skills. I was able to do more than the required which is fully deploying the web application to a cloud server. The learning environment was good and collaborative with people helping each other.

If I am to do this project or a similar project again, I will try to do the web app with a mobile-first approach and try to flesh out all the requirements more in-depth before starting to add more dependencies to my project.

**References**

Future Skills Academy. August 2022. JavaScript.

*Count clicks on links with MySQL and JavaScript*. (n.d.). Stack Overflow. <https://stackoverflow.com/questions/50994256/count-click-on-links-with-mysql-and-javascript>

*Count the number of selections in a multiple-select box*. (n.d.). Stack Overflow. <https://stackoverflow.com/questions/12549770/count-the-number-of-selections-in-a-multiple-select-box>

*CSS Tooltip*. (n.d.). W3Schools Online Web Tutorials. <https://www.w3schools.com/css/css_tooltip.asp>

*How to set input type date's default value to today?* (n.d.). Stack Overflow. <https://stackoverflow.com/questions/6982692/how-to-set-input-type-dates-default-value-to-today#:~:text=var%20today%20%3D%20moment().,way%20to%20solve%20this%20issue>

*- HTML: Hypertext Markup Language | MDN*. (2022, August 10). MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/date>

*The importance of cultural aspects in impact assessment and project development: Reflections from a case study of a hydroelectric dam in Brazil*. (2016, 1). Taylor & Francis. <https://www.tandfonline.com/doi/full/10.1080/14615517.2016.1184501>

Nurullah, M. (2021, August 30). *Create registration and login form in Node.js & MySQL*. CodingStatus. <https://codingstatus.com/create-registration-and-login-form-in-node-js-mysql/>

Peek, S. (2013, August 26). *What is Agile Scrum methodology?* Business News Daily. <https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html>

*What is this popup thing when you Hover over something?* (n.d.). Stack Overflow. <https://stackoverflow.com/questions/65075078/what-is-this-popup-thing-when-you-hover-over-something>

SQL - SELECT from multiple tables with MS SQL server - GeeksforGeeks. (2021, June 14). GeeksforGeeks. <https://www.geeksforgeeks.org/sql-select-from-multiple-tables-with-ms-sql-server/>

*Build a contact form in Node.js with SendGrid*. (2022, January 13). Twilio Blog. <https://www.twilio.com/blog/build-contact-form-node-js-sendgrid>

*Chart.js*. (2022, August 3). Chart.js | Open source HTML5 Charts for your website. <https://www.chartjs.org/docs/latest/>

*Failed to deserialize user out of session*. (n.d.). Stack Overflow. <https://stackoverflow.com/questions/64291646/failed-to-deserialize-user-out-of-session>

Raj, R. (2020, December 29). *Pooling connections in Node.js & MySQL*. Medium. <https://medium.com/dscjssstu/pooling-connections-in-node-js-mysql-9685d5c03c30>

Tribhu, J. (2021, January 24). *Build a login system in Node.js using Passport.js and MySQL*. Medium. <https://jodiss-tri.medium.com/build-a-login-system-in-node-js-using-passport-js-and-mysql-52667cf3cc40>

# Appendix: Source Code

Please open the zip file provided in the submission folder.