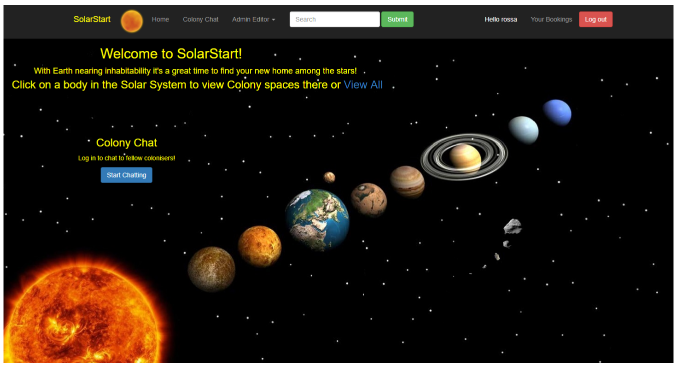
SolarStart

Rossa Heraty Quirke
40204617@napier.ac.uk
Edinburgh Napier University - Advanced Web Technologies (SET09103)

1 Introduction

Humans have colonised the solar system. SolarStart is a site that allows registered users to book places on these colonies and emigrate from Earth. There are multiple ways to navigate the site, the most prominent is via the home page which is a clickable map of the solar system, for example clicking on Mars will lead you to a page displaying all available colony places on Mars, see Appendix B.



Welcome to SolarStart!

With Earth nearing inhabitability it's a great time to find your new home among the stars!

Click on a body in the Solar System to view Colony spaces there or View All

Colony Chat

Log in to chat to fellow colonisers!

Home page with clickable solar system

Users can register a new account, for which they have to provide a valid email address, the site then sends an email to confirm successful registration. Once registered they can log in and make a booking. A booking can be made through entering the amount of places the user wishes to book, on the selected colony, and a 'Citizen Number'. An email confirmation with the cost of the booking and the number of places booked is sent to the email associated with the user. Users can also chat to each other via the Colony Chat page and view all the bookings they have made in the View Bookings page, where they can also cancel a booking. If a user has admin privileges they can create new colony booking advertisements or delete existing ones. Once a colony booking has had all its places booked by users the booking is removed from the site.

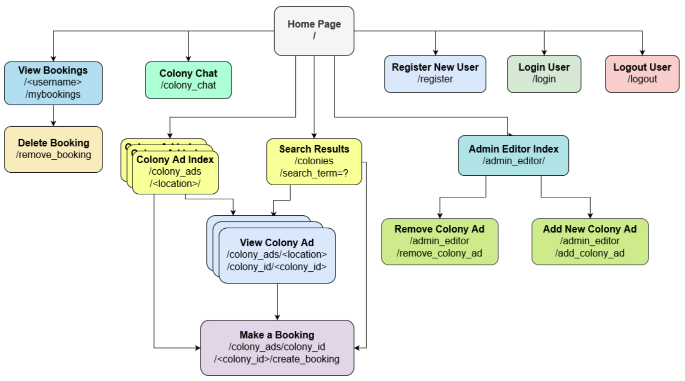
2 Design

As well as the clickable solar system, users navigate the site through the navigation bar at the top of the site's pages, as the navigation bar is present on all pages this allows quick and easy navigation to all parts of the site. On the navigation bar is a search bar that will return available bookings where the name or location match, or are similar to, the search term entered. Bookings can be made either from the search results, location filtered ads, or the individual ad page, see Appendix C. In addition to there are links to register and login/logout, clicking these creates a modal popup allowing for a user to input their details and complete each of these actions, see Appendices E and D. Once a registered user has logged it then now have access to two other pages, these are the Colony Chat and View Bookings pages.

The Colony Chat is a chat page consists of a panel displaying sent messages, user connects, and user disconnects; and a text box and enter button to send their own messages, see Appendix F. Messages are seen by any other user also connected to the page at the current time, allowing multiple users to participate in a site wide conversation. The chat was implemented with the use of Flask's SocketIO and uses information stored in the Flask session to display which users have connected/disconnected and who sent each message.

On the View Bookings page a user can view any bookings they have made, including the number of spaces and total cost. The bookings are presented as a list of panels with the image related to the bookings and information below, see Appendix G. Bookings can also be deleted on this page via the 'Cancel Booking' button, though the site asks the user to confirm again they wish to cancel after the clicking the button.

If a user has administrator privileges then two new pages are available. One to create a new colony booking advertisement and the other to delete existing advertisements. Both pages are forms which require the admin to input information, see Appendices H and I. The 'Add' page requires all details related to the booking, such as image, cost per person, and location. The 'Remove' page requires just the Colony ID, a field by the program, that is displayed on all advertisements.



URL Hierarchy

The app is structured so that a user can quickly navigate to all high pages quickly and easily, accomplished through multiple methods of navigation such as the navigation bar and search bar. Pages are also hyper linked, for example a colony advertisement will link to the index of colony advertisements in the same location, this would be useful in the case of when a term is searched a user wants to explore other colonies in the same location as one of the results. Users can supply sensitive data during the registration and log in process. Both processes require their email and a password. To make the app secure hashing was used on the user's password through the use the Flask Bcrypt library. In the case of a database breach all users' accounts would not be comprised as the data that would allow a malicious actor to log in to their account is encrypted. While the session stores some information about the current logged in user it does not contain their email or password, instead a unique user id is assigned to each user and stored in the session, so neither the email or password can be found through the cache. If the app requires other user information, such the user's email address, it simply queries the database to get the email.

3 Enhancements

If given a chance to improve the app there are a few features I would add and expand upon, ranging from smaller adjustments to new features. The first would be an improvement to the Colony Chat feature. I would expand it so that there are multiple rooms, separated by location. If there was a high number of users on the site having a single chat room for all of them to use is inadequate. By dividing rooms by location users could engage in more relevant and related topics âĂŞ such as the journey to that location or a colony in that location, rather than broadcasting to all users who may wish to discuss other topics. And with a large number of users simultaneously chatting would result in too much clutter in the chat and a breakdown of communication. In addition to this expansion I would add admin privileges so that an admin can mute selected users, in the case that a user is in breach of the code of conduct.

A new feature I would add is a payment process so that users could pay for their booking at the website, instead of, assumedly, arriving at the colony and being stripped of their possession as payment. This feature would require a valid payment method to be supplied, multiple methods should be accepted such as PayPal, credit card, and debit card. Payment would be confirmed before a booking is complete and the confirmation sent after. The feature would require a lot of considerations about the security of this process in order to keep a user's payment information and method secure, and verification that the user is who they say they are and that the payment is processed correctly.

A general improvement to the quality of the app would be more extensive unit testing. Currently there is testing for pages, checking they are functioning correctly by asserting a 200, and a 302 when a redirect is expected, due to the user's logged in or admin status. For example, adding tests to assert that the expected database entry is created when making a booking, and that it is deleted when cancelled by a user would make the app more robust.

4 Critical Evaluation

The app has a wide breath of functionality that allows for a full user experience. It replicates successfully many of the core features of similar booking style websites, and adds a social aspect through the chat. The overall site is uniform in it's style, with familiar aspects such as the navigation bar and the white display panels with the starry background, which makes sure the user does not suddenly feel lost when going to a new page. I feel that the booking process is smooth and intuitive, the path to create a booking is clear, and can be accomplished in only a few clicks starting on the home page, with an email confirmation for a completed booking being sent to the user, an expected feature for modern booking sites, this was achieved through Flask Mail. The app handles registration and login/logout in a compact and effective way through the use of Bootstrap modal forms where a user inputs their data, these forms can be brought up via the navigation bar from any page on the site and dismissed without ever leaving the page so that a user can login when needed and not have to navigation multiple links to login to reach page that requires this. The use of these forms makes the registration and login/logout process very responsive.

On the other hand there are a few aspects of the app I feel do not perform as well as the ones mentioned above. While the Colony Chat accomplishes the goal of allowing users to converse with each other the messages are not persistent across a user's time logged in. If they navigate away from the chat page they will not be able to see any of the messages sent while they were away. And as there is a single chat for the entire app, multiple users conversing at once would quickly become confusing and hard to track.

Another part of the app that falls a bit short is the error handling during the login and registration process. I had the idea of having Flask flash messages sent to appear on the modal form for either login or registration if the user inputted incorrect data. However I found that the flashed message would appear in both modals even if triggered by just one. Instead I opted to send the user to a separate page that handles these errors with flashed messages and a user can login or register from that page. I would have liked to have these processes totally contained within the Bootstrap modal forms but I decided that overall it was a small issue and my time would be better spent in adding new features to the site, though this does mean this aspect falls below the bar in my opinion.

5 Personal Evaluation

There was a lot of creative freedom in this coursework and the first major challenge was to think of concept for my web app that would allow me to effectively showcase what I have learned in a novel way while still being a feasible amount of work. To fully utilise the time assigned for the project I implemented the registration and login/logout process, and encryption for user data, before settling on an idea, as this is generic functionality common to most websites. After further brainstorming and watch I settled on the idea of a booking site for space colonies, I thought a booking style site allowed for a good range of functionality and gave me many sources to look at for inspiration, such as AirBnB and Booking.com. I also thought it was a unique spin on the concept and would motivate me to put more effort in as I am a big fan of science fiction. I feel this approach of implementing generic features, while still considering themes for the site, was a good decision and took pressure off of deciding on an idea, allowing me to eventually come up with something I was keen to create.

Time management was an important part of this coursework, as I had multiple other deadlines due in within a short time from each other I had to time block when I would work on each project and break down this project. In my last coursework I noticed I spent too much time on small graphical interface details instead of taking a step back and implementing new features. This time I created a list of features I wanted in the site and ranked ordered them by importance. This helped me to prioritise and time block my work, as I knew to spend the most time on the most core features I knew I wanted.

Overall I felt I preformed well in the development of my own web app. The work done across this module has been my first experience in web development and the first time utilising JavaScript and CSS. My previous experience in Python was limited and had no knowledge of the Flask micro-framework. From that starting position I am very satisfied with the progress I made and how confident I have become in using these tools. The process of finding new libraries in order to achieve specific features I wanted was also very useful. In conclusion I feel handled the project well and learned skills that will be useful in any future software development, not just for the web.

References

[1] SocketIO documentation, Retrieved from: https://socket.io/get-started/chat SocketIO Chat Tutorial

[2] Tutorialspoint.com, Retrieved from: https://www.tutorialspoint.com/flask/flask-mail.htm

[3] PythonHosted.Org Flask-MailÂű Retrievedfrom:https://pythonhosted.org/flask-mail

[4] W3Schools, Retried from: https://www.w3schools.com/bootstrap/bootstrap_modal.asp Tutorial for Bootstrap Modal Forms

[5] Flask Unit Testing Tutorial https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-vii-unit-testing-legacy

[6] Flask Unit Testing Sessions Documentation http://flask.pocoo.org/docs/1.0/testing/#accessing-and-modifying-sessions

[7] Jinja2 Examples https://realpython.com/primer-on-jinja-templating/#quick-examples

[8] SQL using WHERE LIKE https://www.dofactory.com/sql/where-like

[9] Jinja2 Documentation http://jinja.pocoo.org/

[10] StackOverflow answer to search term in URL https://stackoverflow.com/questions/11774265/how-do-you-get-a-query-string-on-flask

[11] Robinson, K. (2018, August 15) Send Email programmatically with Gmail, Python, and Flask. Retrieved from: https://www.twilio.com/blog/2018/03/send-email-programmatically-with-gmail-python-and-flask.html

[12] Flask-Bcrypt Docs, Retrieved from: https://flask-bcrypt.readthedocs.io/en/latest/

[13] W3Schools HTML tutorial, Retrieved from: https://www.w3schools.com/tags/tag_map.asp Used for solar system clickable map

Appendices

A Acknowledgement of Additional Libraries Used

To achieve the chat feature of my web app I used the Flask SocketIO library which was not covered in the teaching materials for this module. I chose SocketIO as it is a simple and easy to use library to add web socket functionality which a large amount of online resources and documentation, such as those that I cited in my references. SocketIO additionally requires a networking library to run, for which I chose 'eventlet'. The installation process for these is:

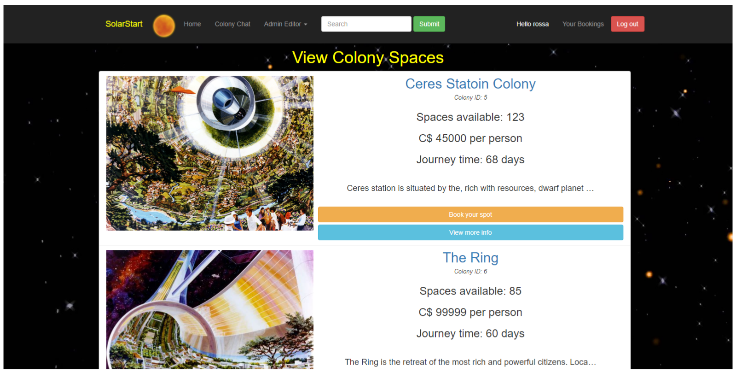
```
$ pip install Flask−SocketIO
$ pip install eventlet
```

I also made use of the Flask Mail library. I felt this would create a fuller experience for the user and so decided to add email functionality, Flask Mail provides an easy framework to send emails encoded with HTML. The installation process is as follows:

```
$ pip install Flask−Mail
```

Lastly the website https://www.image-map.net/ was used to generate the HTMl mappings to create the clickable solar system image on the home page. The image itself was sourced online and edited by myself.

B View Colony Ads for a Location



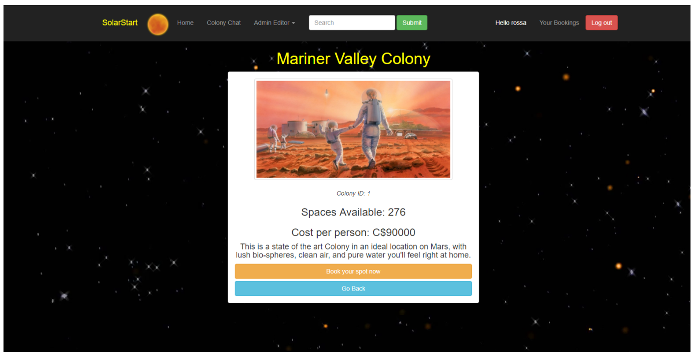
View Colony Spaces

Ceres Statoin Colony
Colony ID: 5
Spaces available: 123
C$ 45000 per person
Journey time: 68 days
Ceres station is situated by the, rich with resources, dwarf planet ...

The Ring
Colony ID: 6
Spaces available: 85
C$ 99999 per person
Journey time: 60 days
The Ring is the retreat of the most rich and powerful citizens. Loca...

Page showing Colony Ads for 'The Belt'

C View Colony Information



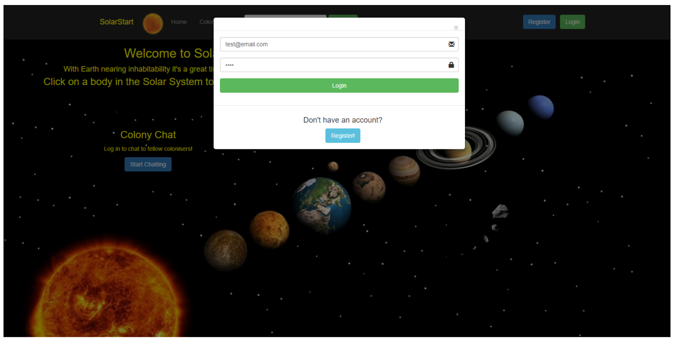
Spaces Available: 276

Cost per person: C$90000

This is a state of the art Colony in an ideal location on Mars, with lush bio-spheres, clean air, and pure water you'll feel right at home.

Page showing an individual Colony Booking Ad

D Login Bootstrap Modal Form

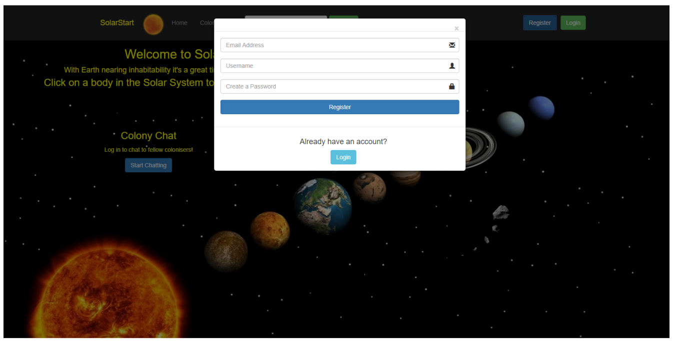


With Earth nearing inhabitability it's a great ti

Click on a body in the Solar System to

Login Bootstrap Modal form with example email and password

E Register Modal Form

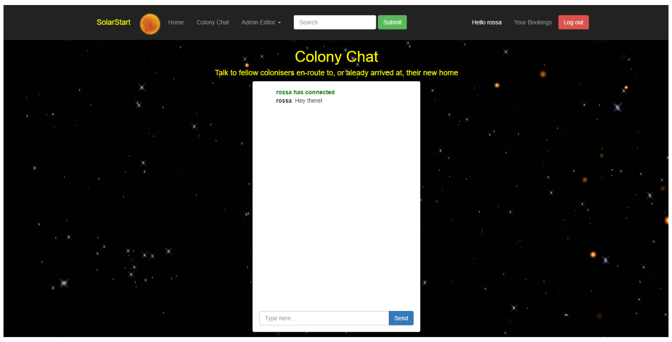


With Earth nearing inhabitability it's a great ti

Click on a body in the Solar System to

Register Bootstrap Modal form

F Colony Chat Page



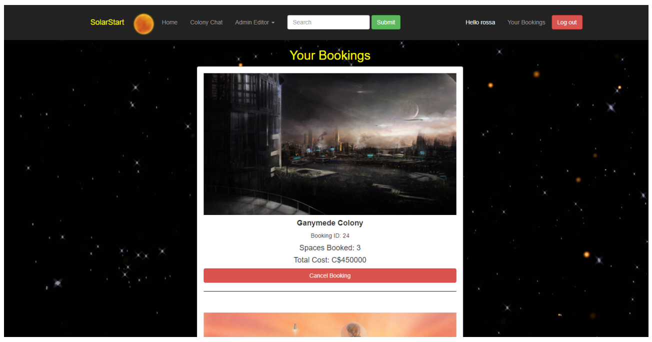
Talk to fellow colonisers en-route to, or already arrived at, their new home

rossa has connected

rossa Hey there!

Colony Chat with a user connected and message sent

G View Bookings Page



Ganymede Colony

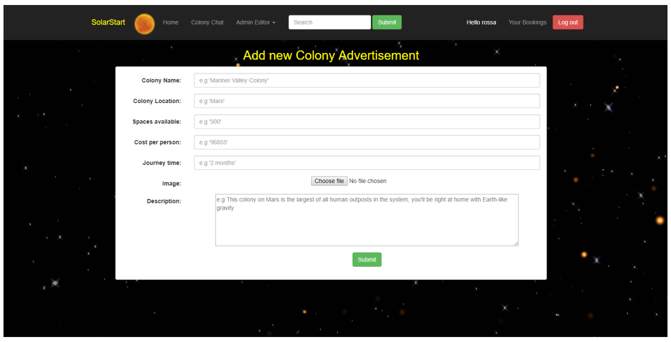
Booking ID: 24

Spaces Booked: 3

Total Cost: C$450000

Page showing a user's bookings with a booking for the Ganymede Colony

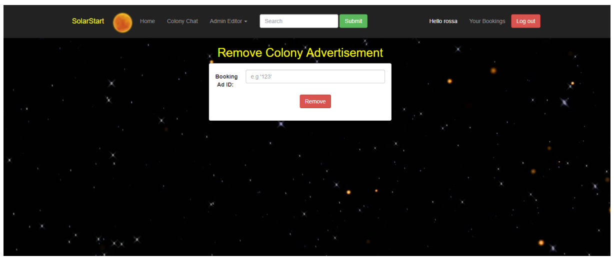
H Add New Colony Ad Page



Colony Name:	e.g 'Mariner Valley Colony'
Colony Location:	e.g 'Mars'
Spaces available:	e.g '500'
Cost per person:	e.g '96000'
Journey time:	e.g '2 months'
Image:	Choose file No file chosen
Description:	e.g This colony on Mars is the largest of all human outposts in the system, you'll be right at home with Earth-like gravity

Create new Colony Ad Admin tool

I Remove Colony Ad



Booking Ad ID: e.g '123'

Remove Colony Ad Admin Tool