School of Computer Science and Engineering

UNSW SYDNEY

COMP 3900

Information Technology Project

2022, T3

Team Name: UNeverSleepWell

Project Title: UNeverSleepWell

Team Member:

MINXIN SHI(SCRUM MASTER)	**z5286095@ad.unsw.edu.au**
YIFEI YUE(DEVELOPER)	**z5392319@ad.unsw.edu.au**
FENGBO JIA(DEVELOPER)	**z5265975@ad.unsw.edu.au**
RUIHE CHANG(DEVELOPER)	**z5278982@ad.unsw.edu.au**

Submit time: 18/11/2022

Content

Overview

On the home page of this project, we made a moving snowflake that can be controlled by the mouse. Five buttons are present on the page at once, including this button for testing and two relating to time settings. Because you don't have to wait for real-world time to pass while testing software. The final two options allow you to quickly clear the database system's data and tokens for testing purposes. It should be noted that showing the viable activities within a month needs the existing time, thus if you don't select any time when you first access the page, an error will be displayed on the home page. Make sure to set the time before navigating to our main interface because if it isn't, the system won't be able to query the current time and will have trouble recommending all conceivable activities within a month.

The project needed to find a happy medium between the page's aesthetics and the prominence of its functions. Thus, we just have snowfall on the main page and confetti on the events page as special effects. On other pages, we focused on making the buttons stand out by giving them depth rather than using any fancy effects to draw attention to the functionality.

We employ the top bar and side navigation to provide seamless transitions between sections of the host and user interface. It's always right there in the interface's sidebar, no matter which one you're using. Also, the top bar serves as a fast escape button if you ever need to logout.

Background

Nowadays, the internet is developing rapidly and has become an essential part of people's lives and an important carrier of human civilization's communication and development. In recent years, people have been more inclined to online shopping, book tickets, and so on because of the epidemic's impact. Take concerts as an example; online ticket booking significantly reduces people queuing up to buy tickets and squeezing their heads in line, saving most people's time. Organizers also can spend less money and effort on access to sell tickets. The benefits of the internet are clear to us briefly. However, there are also many convenient booking sites on the internet. For instance, we analyzed two booking websites, TicketMaster and EventBrite, and learned how convenient online ticketing is. These websites saved lots of people's time and made the events more profitable. Furthermore, these websites look nice and clean with great functionality, but they all have three main problems. A brief look at these sites reveals the issues they share positivity, intelligence, and user-friendliness:

Positivity: There are barely any interactions between event organizers and participants, and participants cannot make practical suggestions and ideas to the organizers. We felt it was essential to create interaction and establish a monitoring model where participants must only comment after participating in the event and rate the event on a scale of 0-10, with malicious or inappropriate comments being monitored and removed before they were sent out. This way, we will create a suitable interaction mechanism between the organizer and the participants. The organizer can make changes or upgrade the event from the participants' suggestions to lay the foundation for the next event. Participant scoring of the event will lead to the second problem we want to solve: intelligence.

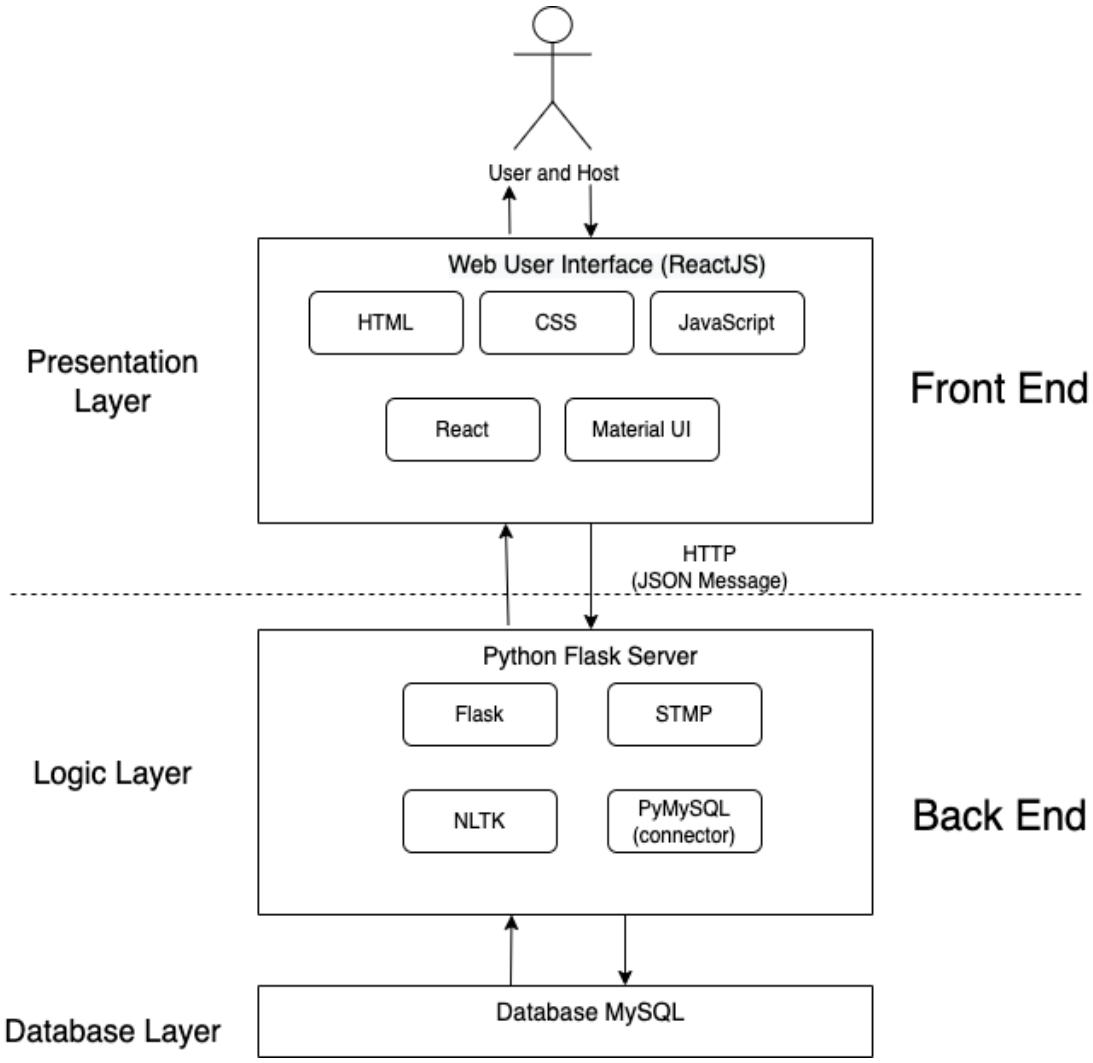
Intelligence: To solve the intelligence problem, we will use the participants' ratings of events to perform machine learning and algorithmic deduction to better target the different preferences of each user. For example, if a customer likes opera and has given good ratings to the opera events they have attended, then our system will intelligently recommend more opera-type events to that user. It avoids the users' need to spend more time finding their favorite type of activities in many activities.

User-friendliness: Most ticketing sites are now non-refundable, and no refunds will be issued once sold. We have improved on this, and we accept full refunds. However, buyers must return the tickets seven days before the event starts because we want to make sure that the event organizer maximizes the benefits and cannot lose the benefits that should be there because of a temporary refund. Moreover, most importantly, we only have one presale to make the event fair.

Our website will have most of the functions other booking sites should have search, view, book, cancel, subscribe, review, etc. Including many kinds of events make the website platform more diverse. And give us new features to add rating, subscription, monitoring mechanism, better recommendation system, correcting the shortcomings of existing booking sites, making our website more comprehensive and practical. It is also more than a booking site; it is a platform for event organizers and event lovers to meet and communicate with each other. Users can follow their favorite organizers and attend all their events. Organizers can also use this platform to show their ability and get more people's likes and recognition.

We look forward to hearing from you and providing us with your valuable feedback.

System Architecture



The system architecture can be divided into three main layers:

• Presentation layer

This layer displays the data to the users in the form of an interface. Under the browser/server-based architecture designs, the user further sends the data requirements to the browser and receives the results as per the management needs. Code interacts with the back-end API by sending a request. The front-end code receives the data in these responses.(Li, 2017)

• Logic layer

The logic layer or the business layer is the core of the application. It is responsible for the business logic of the application. The business layer code runs on the server. It receives requests from the front-end, communicates with the data layer to store the

data and retrieve the results according to the database rules, and returns the results to the front-end in (possibly) JSON format. (*Architecture of Business Layer Working with Entity Framework – The Reformed Programmer*, n.d.)

• Database layer

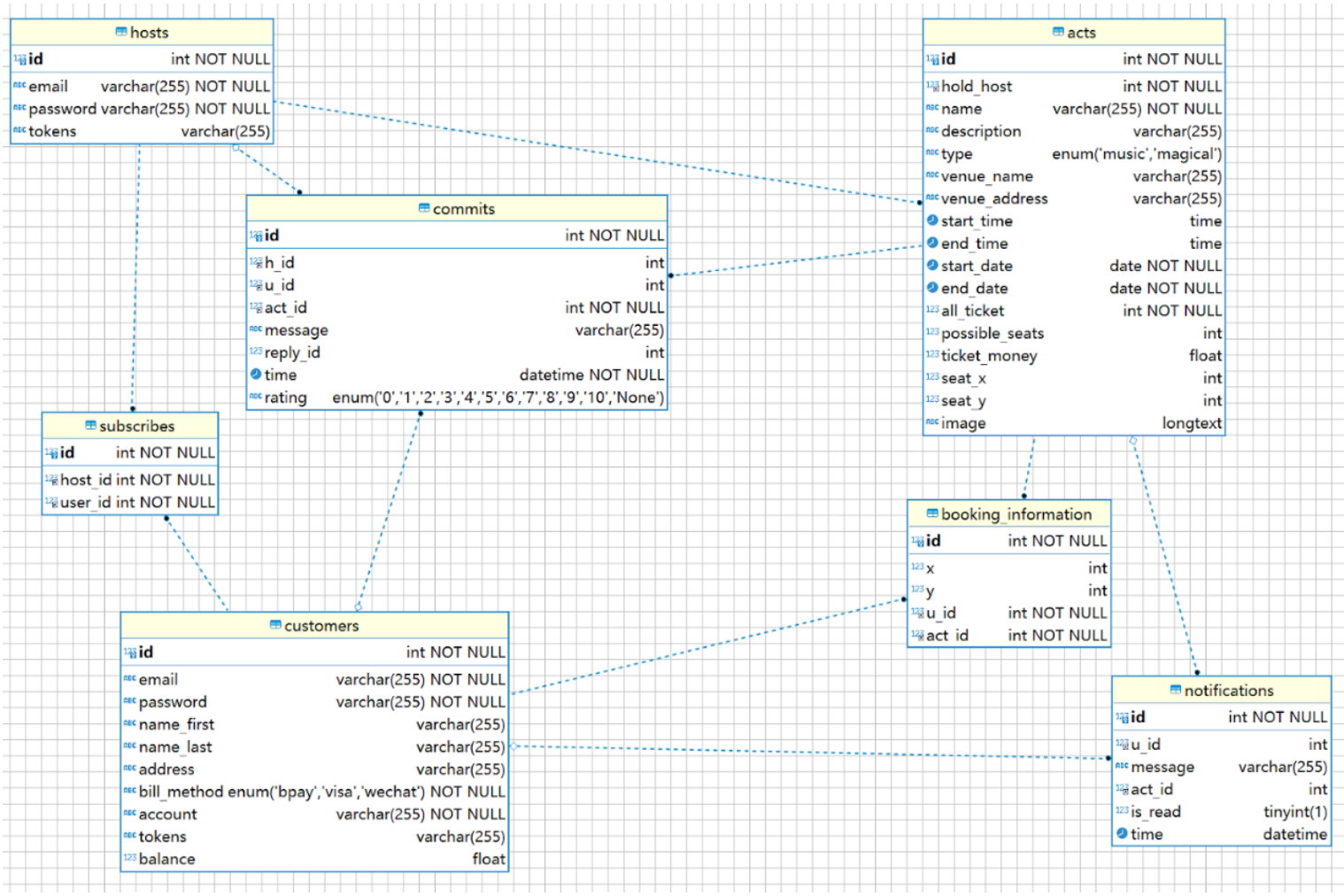
The Database layer consists of technologies that give Web the ability to store and retrieve data. Databases organize data into tables and create relationships between the data in those tables so that a Web can look-up and analyze data in very dynamic and powerful ways. (*Database Layer*, n.d.)

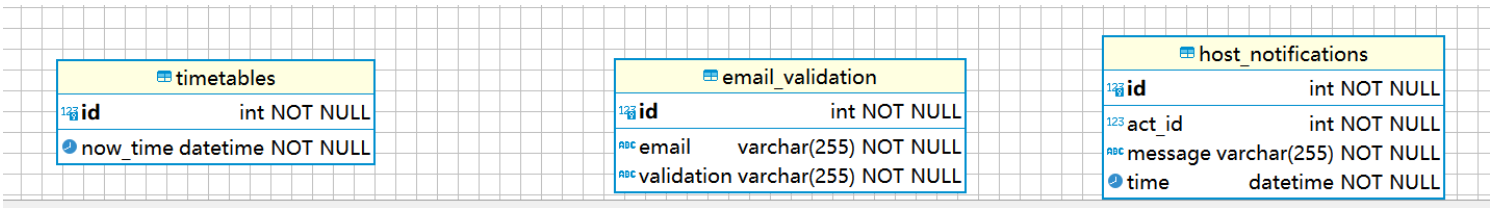
Presentation layer loosely refers to the layer which is responsible for somehow displaying the data for the users. Web users interact with components of the presentation layer to select actions that initiate calls to the underlying layers. Presentation layer components make calls to the logic layer, which in turn sends requests to the database layer (*What Is Three-Tier Architecture - Singapore | IBM*, n.d.)

Used programming language:

SQL, Python, HTML, CSS and JavaScript.

ER Diagram for database





Description of the functionalities

Fundamental functionalities:

Our project completes all the functionalities required for Event Management System. Here we listed all the fundamental functionalities with their relative functions.

1. **Hosts to register and login using username and password credentials.**

   Relative functions:
   Host register -- fundamental
   Host login -- fundamental
   Host logout -- fundamental

2. **Hosts, once registered, to advertise a new event**

   Relative functions:
   Host new activity – fundamental
   Host list activities – fundamental

3. **Prospective customers (anyone in the public) to view a list of events coming up in the next month that have not sold out.**

   Relative functions:
   List all available activities

4. **Prospective customers to register their details, including payment details, to become customers and be able to login.**

   Relative functions:
   User register – fundamental
   User login -- fundamental
   User logout – fundamental

5. **Customers, once registered, to book (purchase) a ticket for a listed event when tickets are available. During the booking process, customers must be able to select an available seat for the event they are booking.**

   Relative functions:
   User add balance – fundamental
   Activity Detail -- fundamental
   User book activity – fundamental
   See Booking Seats -- fundamental
   User list activities – fundamental

6. **Hosts to broadcast a message to all customers that have booked tickets to a given event**

   Relative functions:

   Host broadcast notification -- fundamental

   User all notifications -- fundamental

   User delete notification – fundamental

7. **Hosts to cancel an event that has not started, which should cancel all event bookings, broadcast a cancellation message to all customers with a booking, and refund booking costs to customers.**

   Relative functions:

   Host cancel activity -- fundamental

8. **Customers to cancel their booking if the event is scheduled to occur at least 7 days into the future, and this should refund customer booking costs and free up the ticket spot for someone else.**

   Relative functions:

   User cancel activity – fundamental

9. **Customers, after attending an even they have booked, to leave a review for that even (one review per customer).**

   Relative functions:

   Add Commit/Review– fundamental

10. **Hosts to reply to reviews that have been left by customers for their events.**

    Relative functions:

    Reply Commit/Review– fundamental

11. **Customers and prospective customers to read reviews, as well as host replies to reviews, for all events.**

    Relative functions:

    Activity Detail -- fundamental

12. **Customers to search for events by event title, description, and/or type.**

    Relative functions:

    Search Activity– fundamental

13. **Event recommendations to customers based on the event types and hosts booked in the past. Also provide recommendations to events that have a description, which seems "similar" to descriptions for events that have been booked in the past.**
    Relative functions:
        User Recommendations – fundamental

Novel functionalities:

We also add more novel functionalities. Here we listed all the novel functionalities with their relative functions.

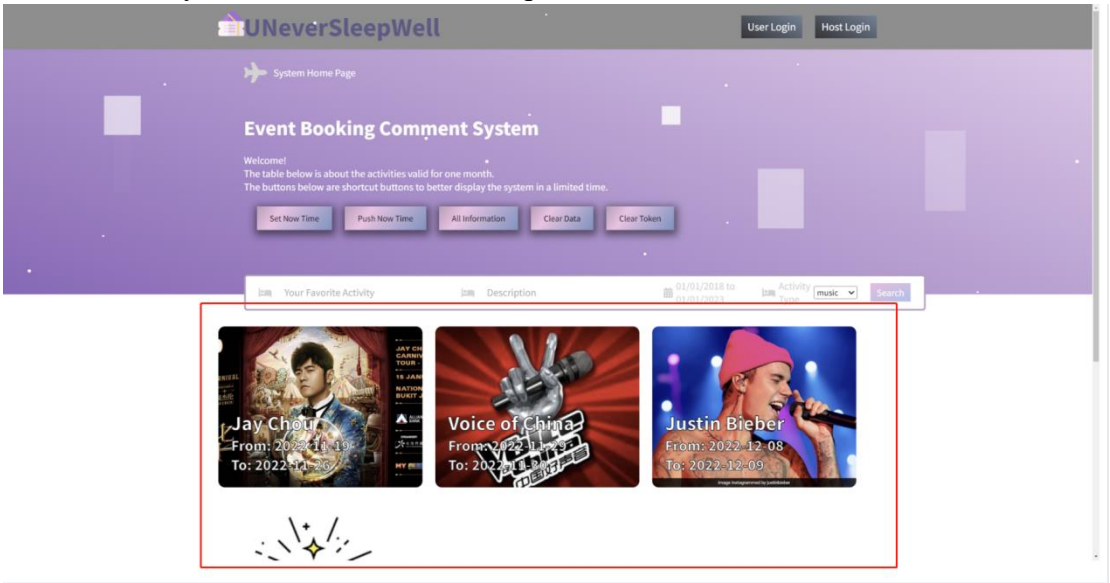
1. **Host and customers can reset password if online and know old password. However, if they forget password, then they need to use email validation to get a validation code and use the correct code to reset password.**
   Relative functions:
   Host reset password – novel
   User reset password -- novel
   Host forget password – novel
   User forget password – novel
2. **User can check their own detail and other users' detail. Also, through their own detail page, they can update their detail and edit their payment account.**
   Relative functions:
   User detail—novel
   User detail update-- novel
   User edit account – novel
3. **User can check hosts 'detail and choose to book a host to be a fan. A host can check all their fans.**
   Relative functions:
   Host detail – novel
   User subscribe host – novel
   Host fan-- novel
4. **Host and user can also edit and reply to other commits, even they can delete their replies or commit/reviews**
   Relative functions:
   Reply commit/review -- novel
   Edit commit/review -- novel
   Remove commit/review -- novel
5. **Chatting system has DFA language filter algorithm to filter all the bad word.**
   Relative functions:
   Commit/Review Filter -- novel

Clarify all the functionalities in detail and by step

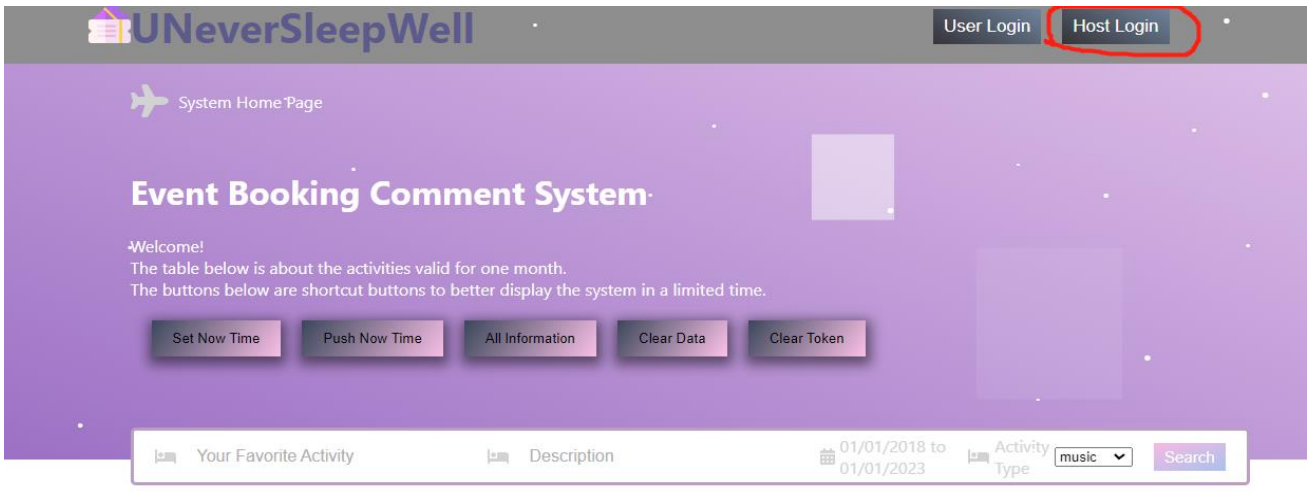
Here we clarify all the fundamental and novel functionalities' relative functions in detail and by step.

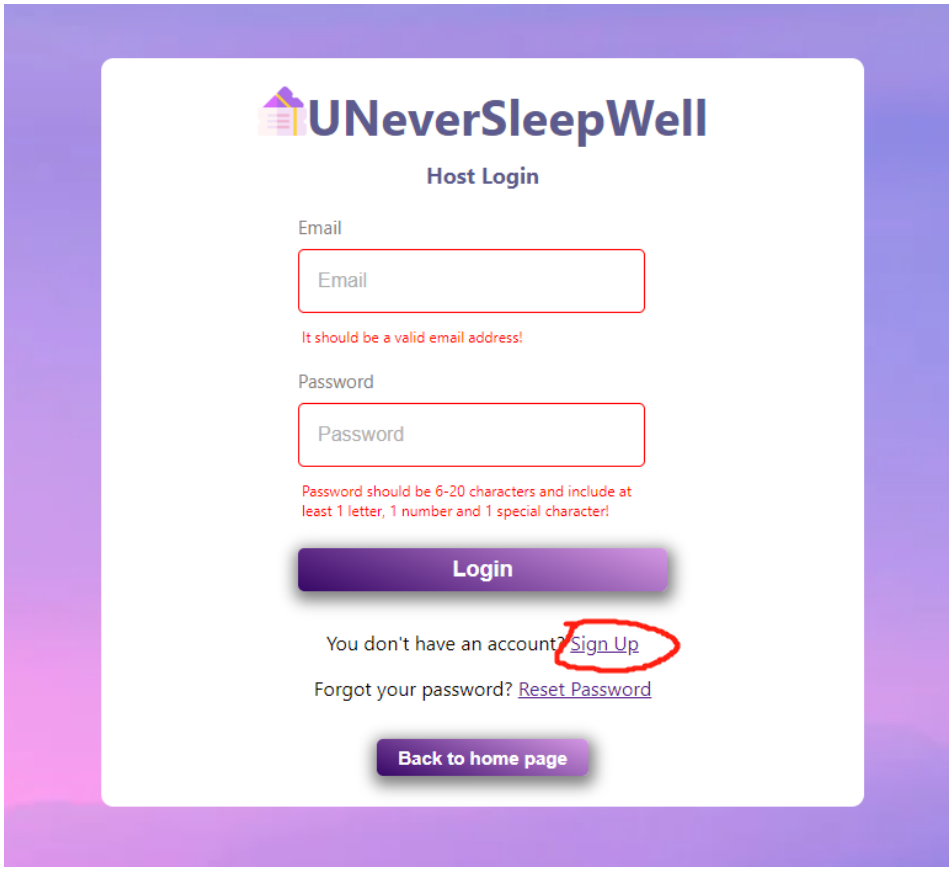
List all available activities – Fundamental functionalities 3
Whether you are a user or a prospective customer, you will get a list of available events coming up in the next month that the tickets have not sold out.

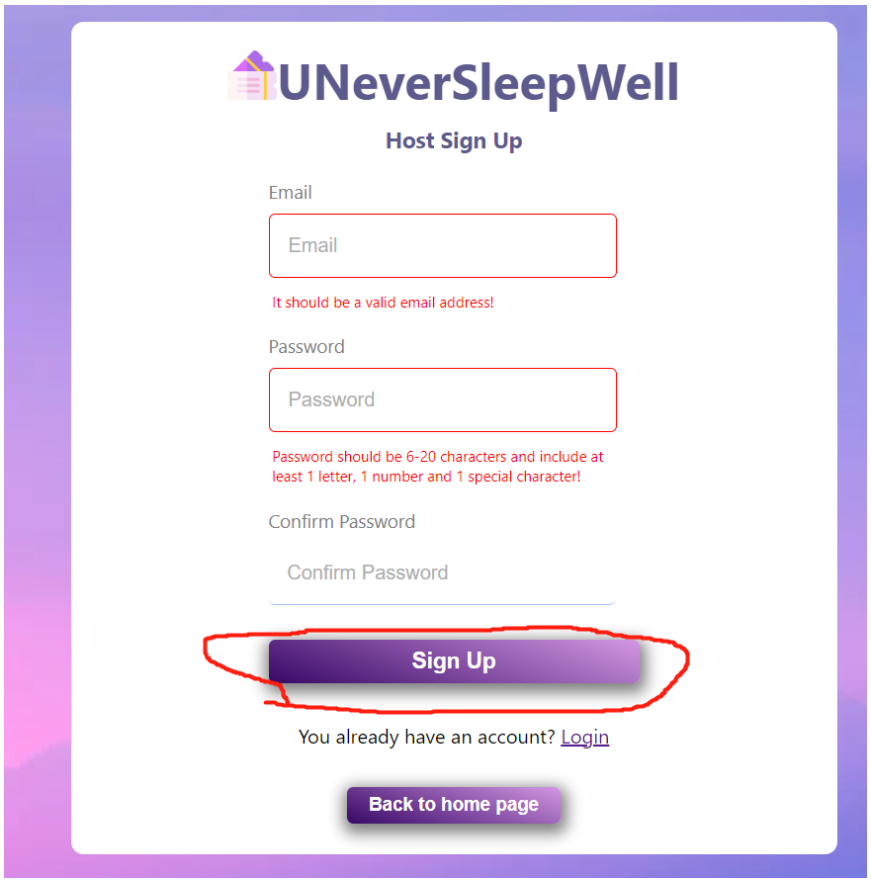
The back end should check all the activities, list out activities that are coming up in the next 30 days, and then from this list, pick out the activities that have not sold out.



Host register – [Fundamental functionalities 1]
To register as a host, need to provide a valid email as a username and a password. When the signup information is sent to the backend, the backend would first check whether the email format is valid and check through the database whether the email has already been used to sign up. If both do not fail, a host should be signed up successfully and directly login to the host home page.

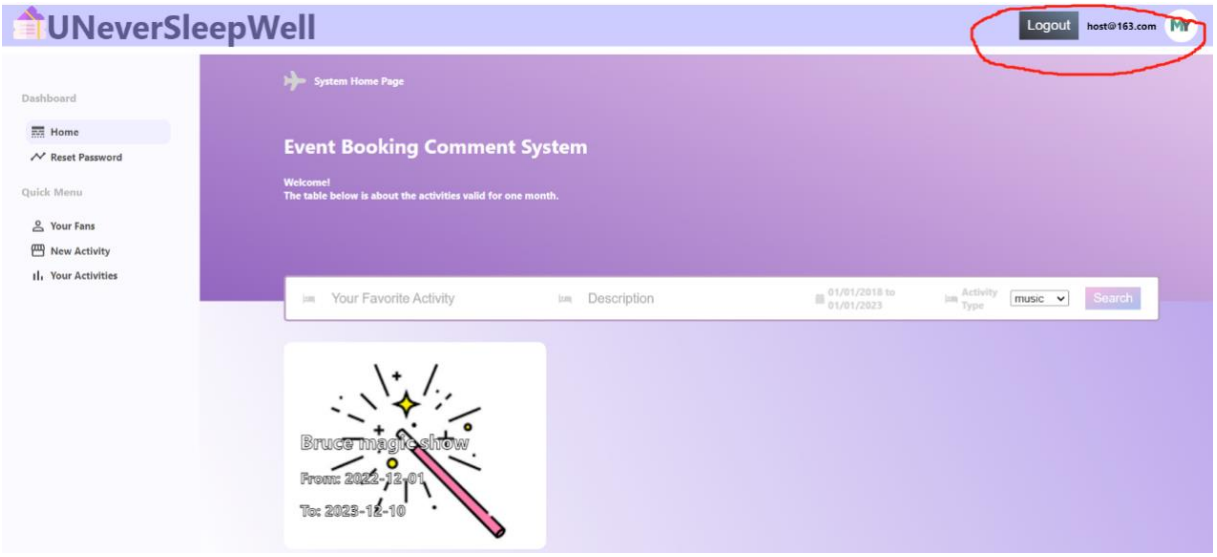



UNeverSleepWell
Host Login
Email
Email
It should be a valid email address!
Password
Password
Password should be 6-20 characters and include at least 1 letter, 1 number and 1 special character!
Login
You don't have an account? Sign Up
Forgot your password? Reset Password
Back to home page


UNeverSleepWell
Host Sign Up
Email
Email
It should be a valid email address!
Password
Password
Password should be 6-20 characters and include at least 1 letter, 1 number and 1 special character!
Confirm Password
Confirm Password
Sign Up
You already have an account? Login
Back to home page

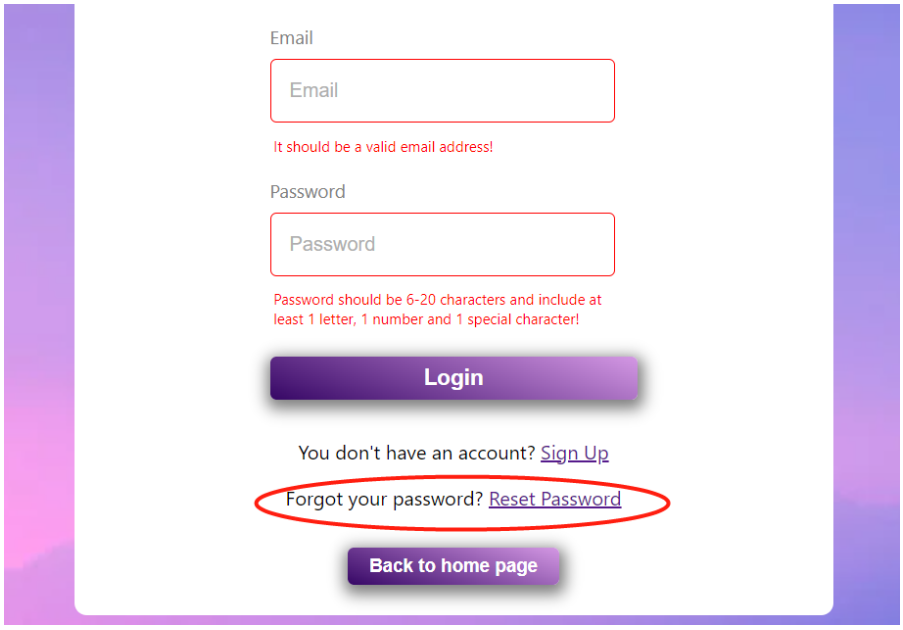
Host logout – [Fundamental functionalities 1]

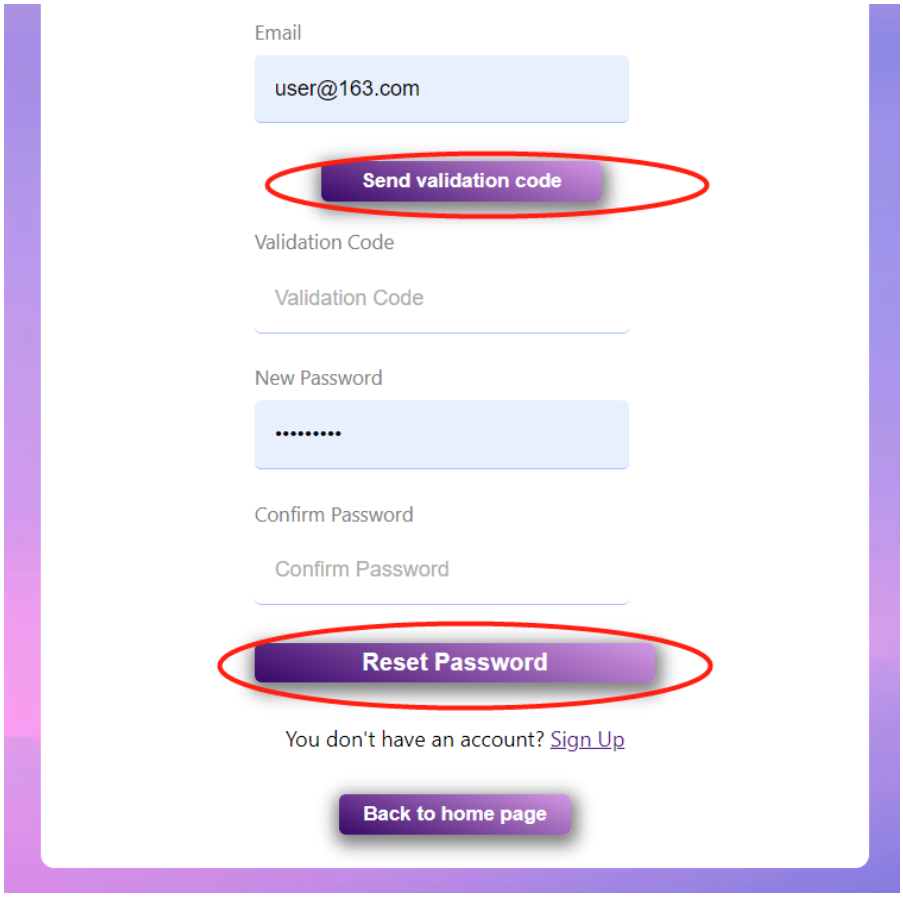
In host home page, by clicking the host logout button, the backend would check through the database whether the host is online by hid and token. If the host is online, then they should be logout successfully and redirected to the login page, with token been cleared.



Host/User forget password – Novel functionalities 1

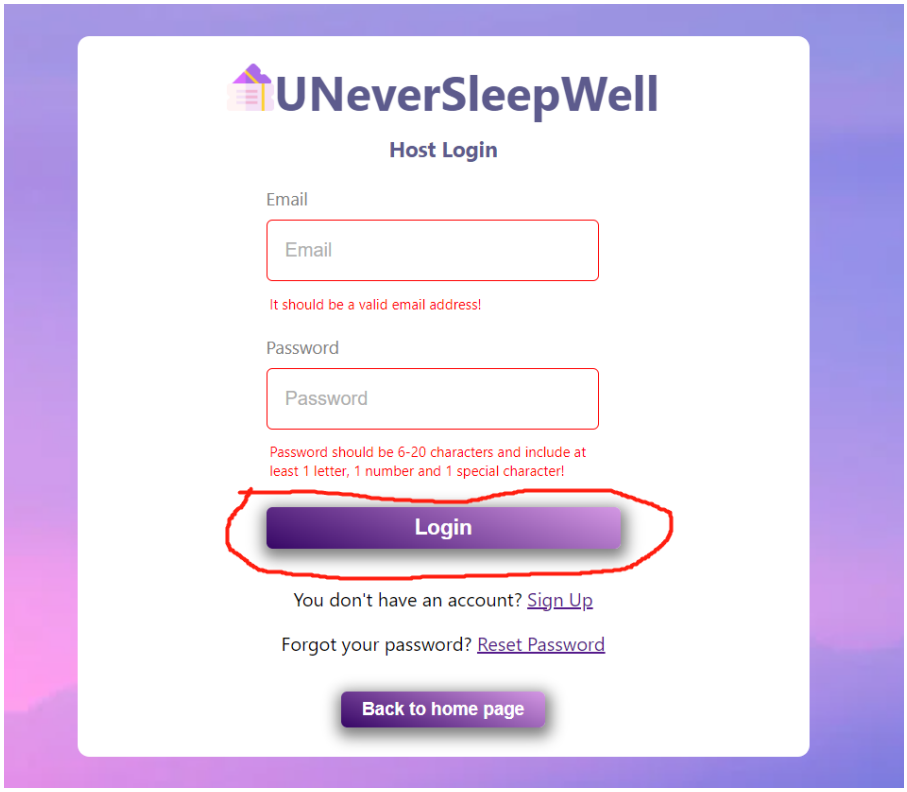
By clicking forget password, host/user should first send an email validation code to their email address. The code is six long random character. After receiving and type the correct validation code, the host/user can reset a new password to their account.





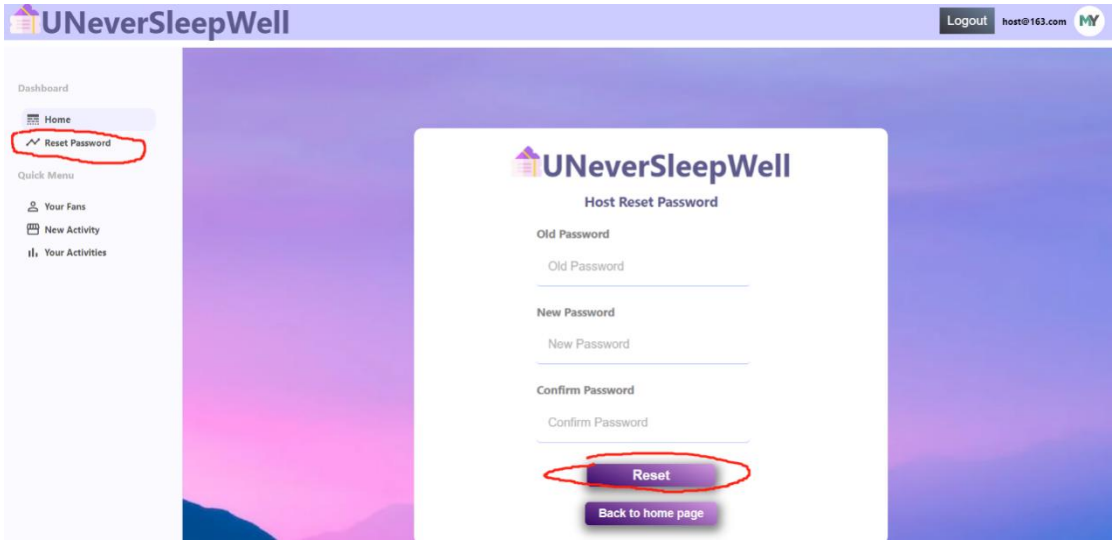
Host login – Fundamental functionalities 1

log in by using a username and password. When the login information is sent to the backend, the backend would check through the database whether the username is a registered host, and then whether the password is matched. If both passes, then a host will be login successfully and be directed to the host home page, with matched hid and token been generated on database.



Host reset password – Novel functionalities 1

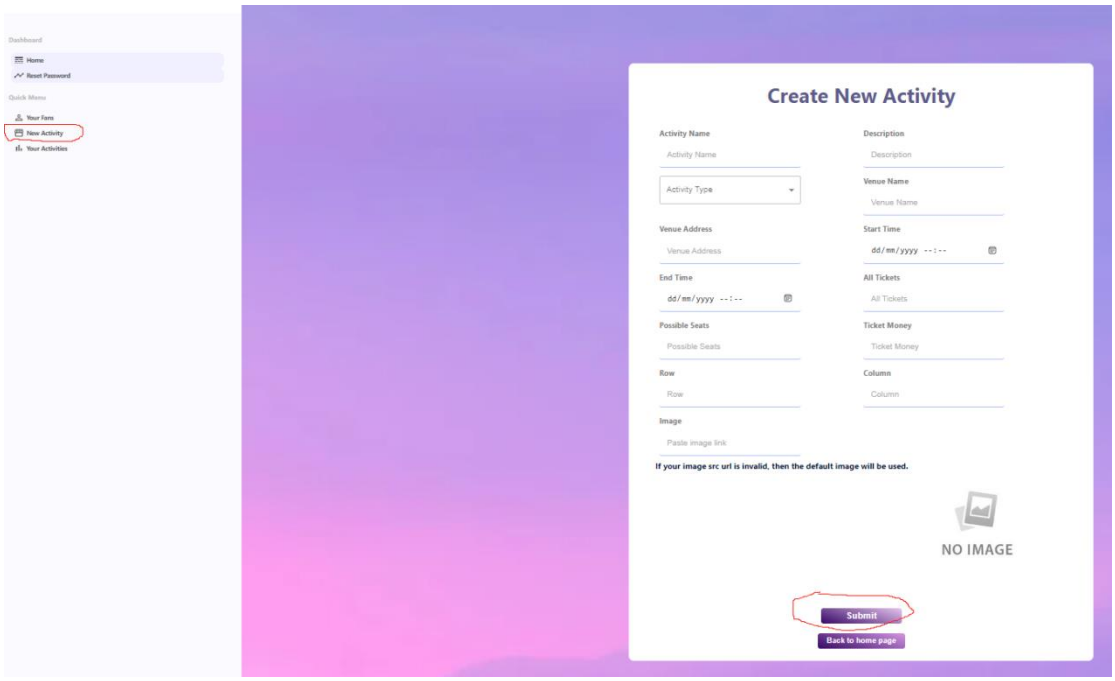
By clicking reset password, the host should be online, the old password should be correct, and the new password should have a valid format. If all satisfy, then the host's password should be reset.



Host new activity – Fundamental functionalities 2

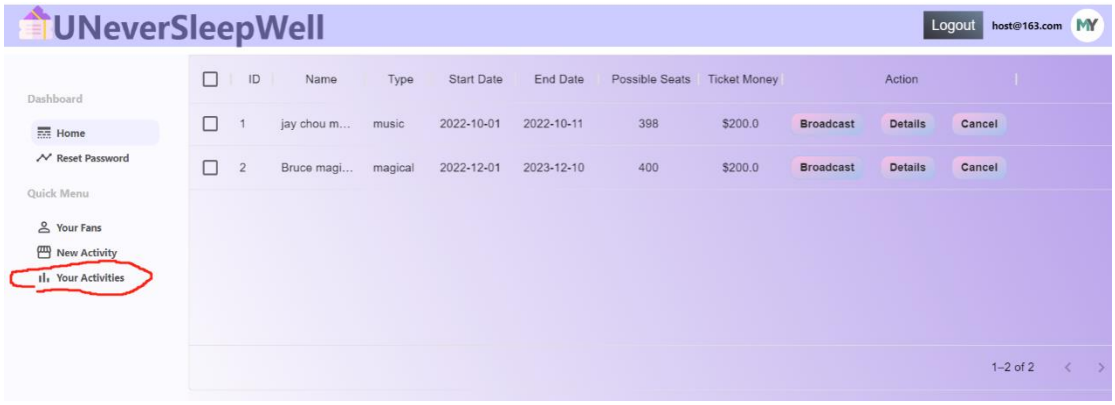
By clicking New Activity, to create a new activity, a host should provide:
activity name, activity description, activity type, venue name, venue address, start time, end time, start date, end date, ticket number, possible seats, ticket money, the row number of seats, column number of seats, and an image link as the activity background image.

The backend would check whether the host is online by hid and token, whether the start date and start time are after time now, and whether end date and end time are after start date and start time. If all satisfy, then create the new activity according to the activity information.



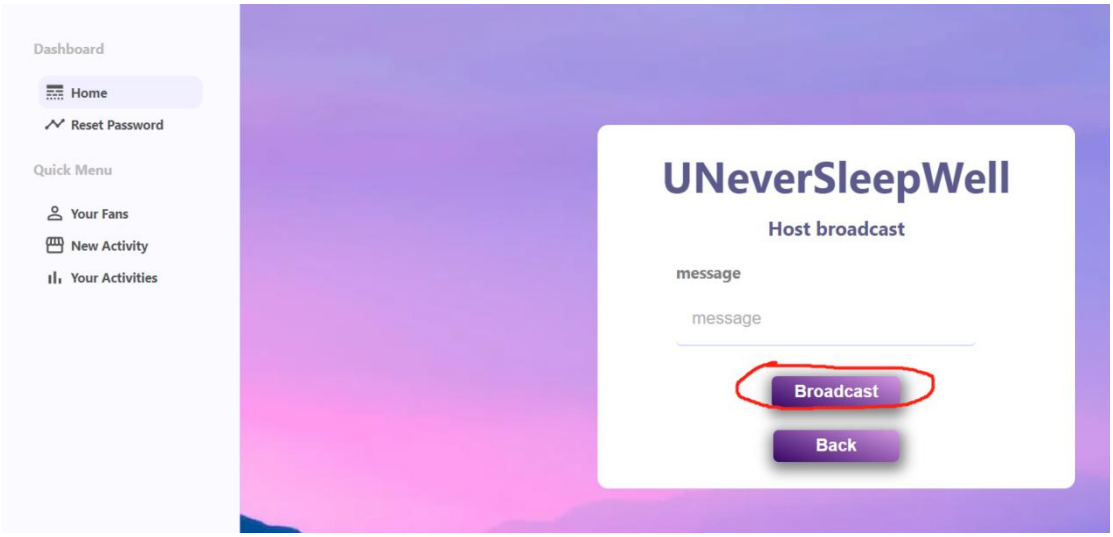
Host list activities – Fundamental functionalities 2

By clicking Your Activities, the host should be online and all the activities they created before will be listed here. They can choose to broadcast, check activity detail, or cancel their activities.



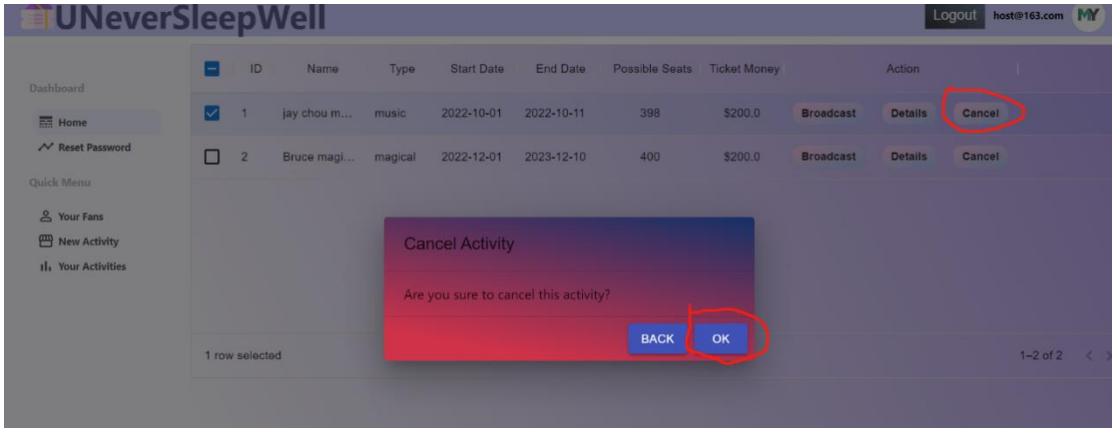
Host broadcast notification – Fundamental functionalities 6

Choosing an activity, clicking broadcast notification, and entering a message, if the host is online, then the message would be broadcast as a notification to all customers within the activity.



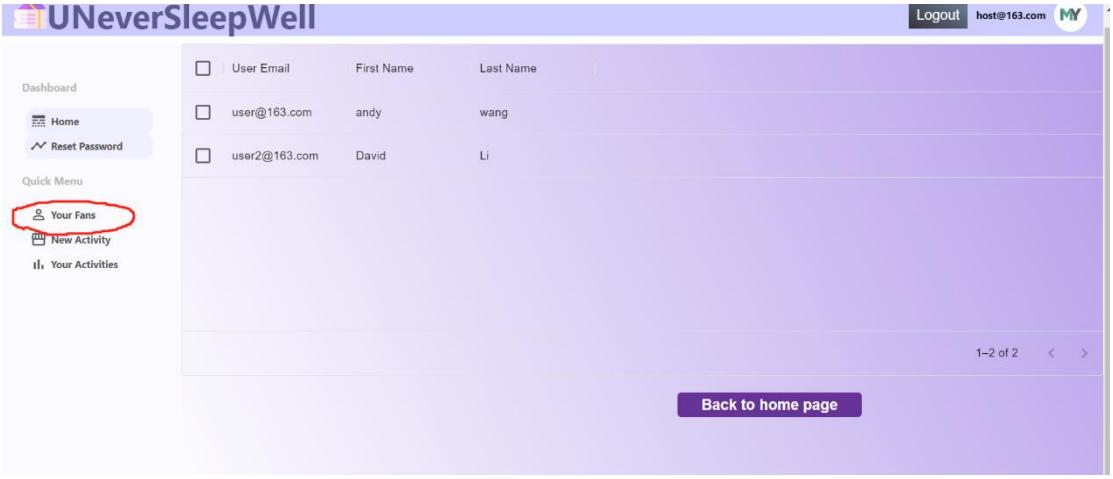
Host cancel activity – Fundamental functionalities 7

When canceling an activity, the backend would check whether the host is online and whether the activity has already begun. If both passes, then the activity should be cancelled, and all customers within the activity would be formed a notification their money would be refunded to their account.



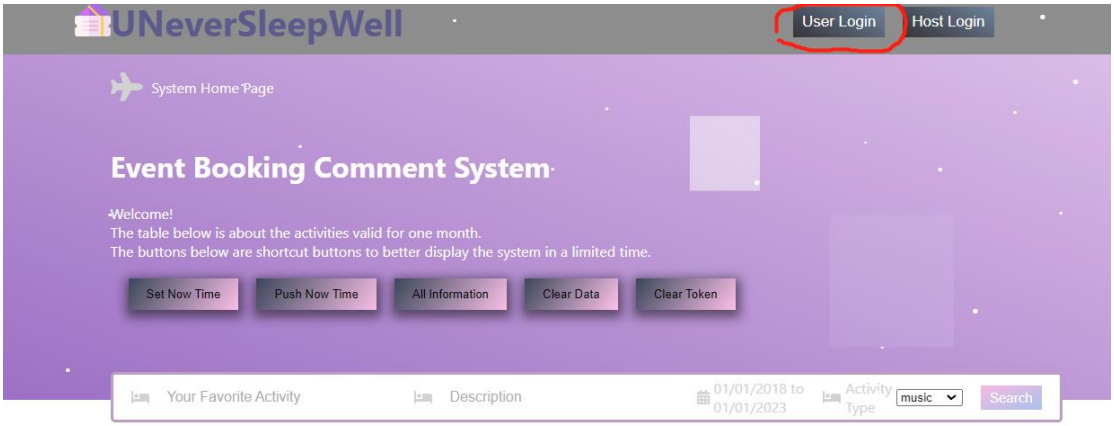
Host fan—Novel functionalities 3

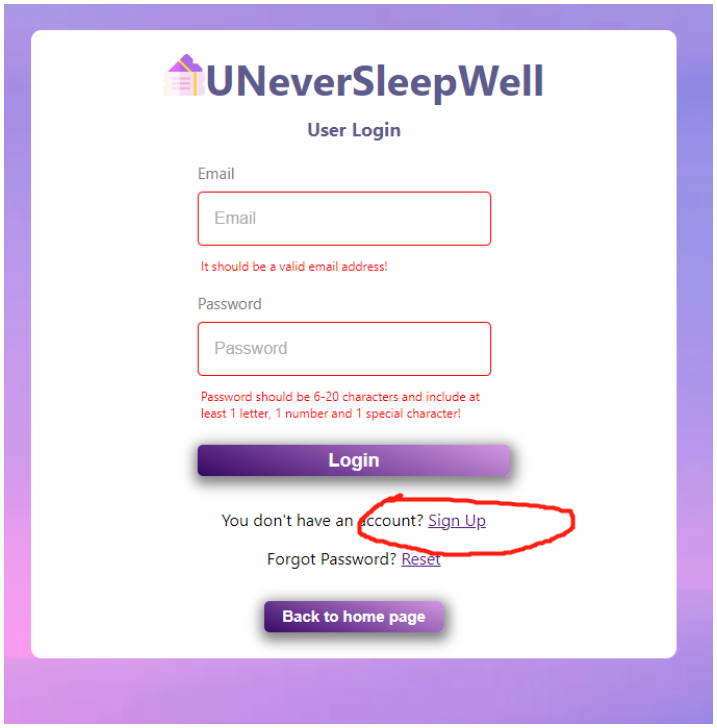
By clicking Your Fans, the host should be online and all the users who subscribed them will be listed here with their email and name.

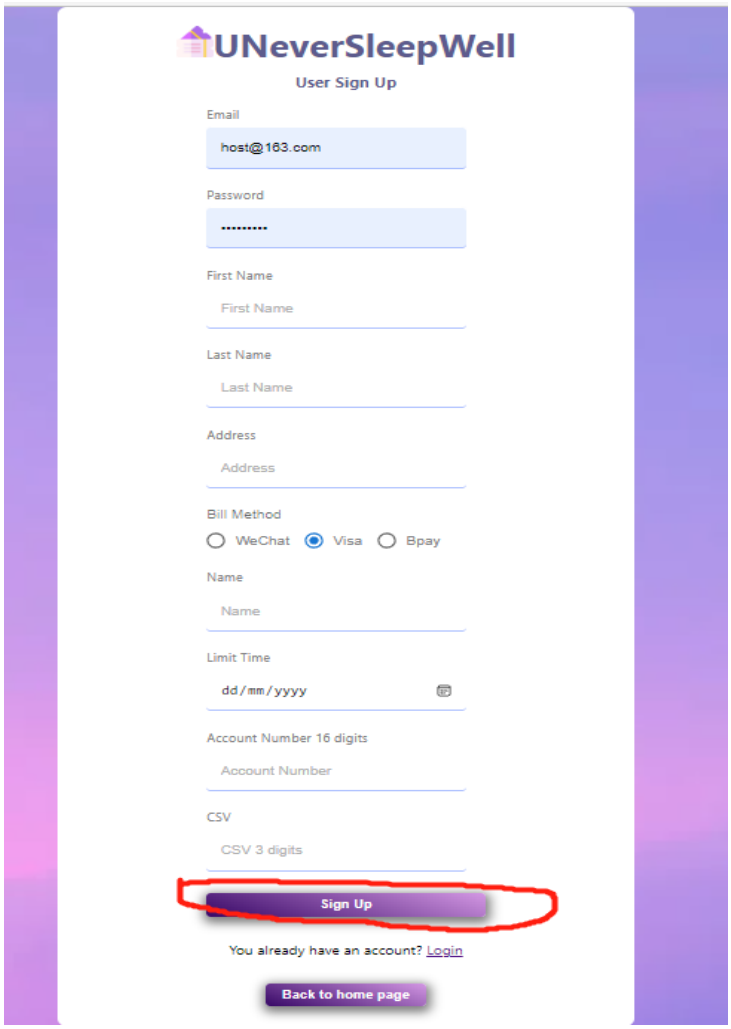


User register – Fundamental functionalities 4

To register as a user, need to provide a valid email as username, a password, first name, last name, address, bill method, and an account. When the signup information is sent to backend, backend would first check whether the email format is valid and check through database whether the email has already been used to sign up. If both do not fail, a user should be signed up successfully and login directly to user home page, and a notification would be sent to the user.

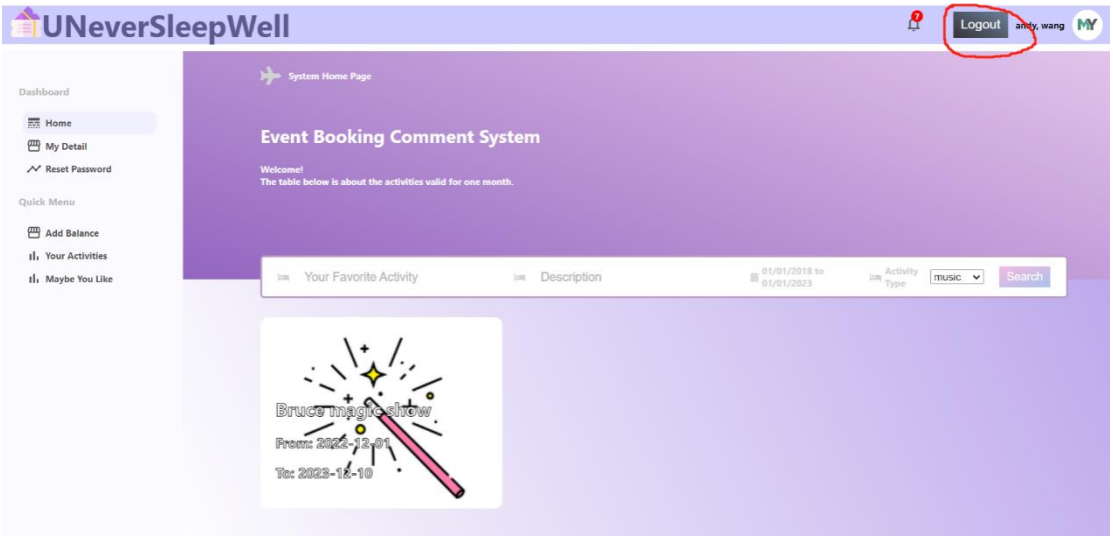



UNeverSleepWell
User Login
Email
Email
It should be a valid email address!
Password
Password
Password should be 6-20 characters and include at least 1 letter, 1 number and 1 special character!
Login
You don't have an account? Sign Up
Forgot Password? Reset
Back to home page


UNeverSleepWell
User Sign Up
Email
host@163.com
Password
First Name
First Name
Last Name
Last Name
Address
Address
Bill Method
WeChat
Visa
Bpay
Name
Name
Limit Time
dd/mm/yyyy
Account Number 16 digits
Account Number
CSV
CSV 3 digits
Sign Up
You already have an account? Login
Back to home page

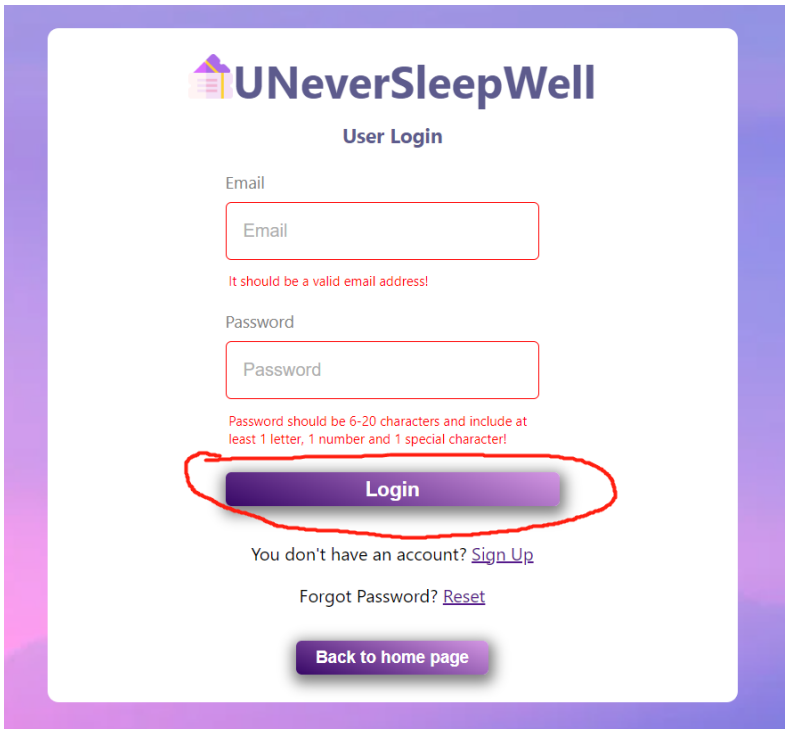
User logout – [Fundamental functionalities 4]

By clicking the user logout button, backend would check through the database whether the user is online. If the user is online, then they should be logout successfully and be redirected to the login page.



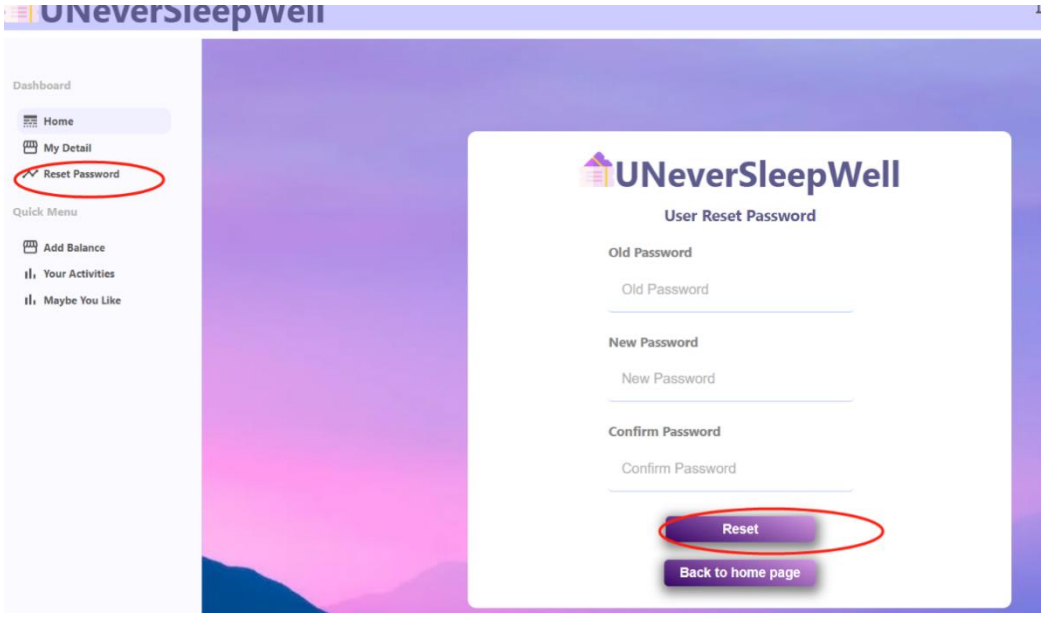
User login – Fundamental functionalities 4

If a user has been signed up, can login by using username and password. When the login information is sent to backend, backend would check through database whether the username is a registered user, and then whether the password is matched. If both passes, then a user will be login successfully and be directed to the user page.

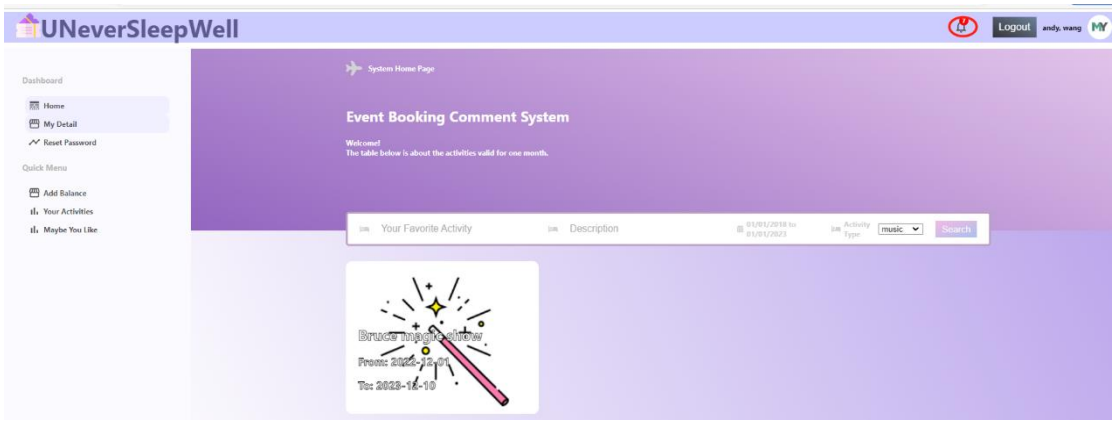


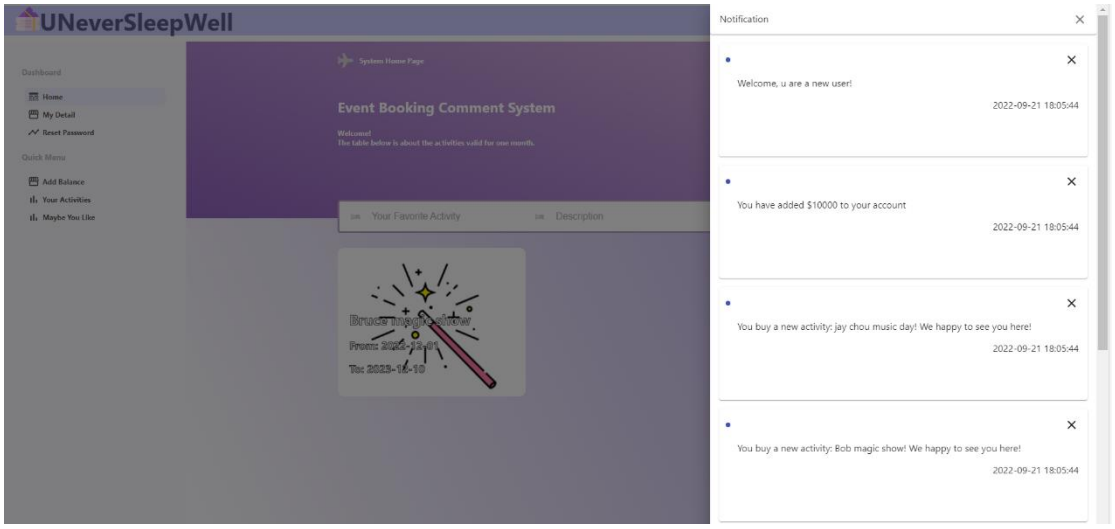
User reset password – Novel functionalities 1

By clicking reset password, the user should be online, the old password should be correct, and the new password should have a valid format. If all satisfy, then the user's password should be reset.



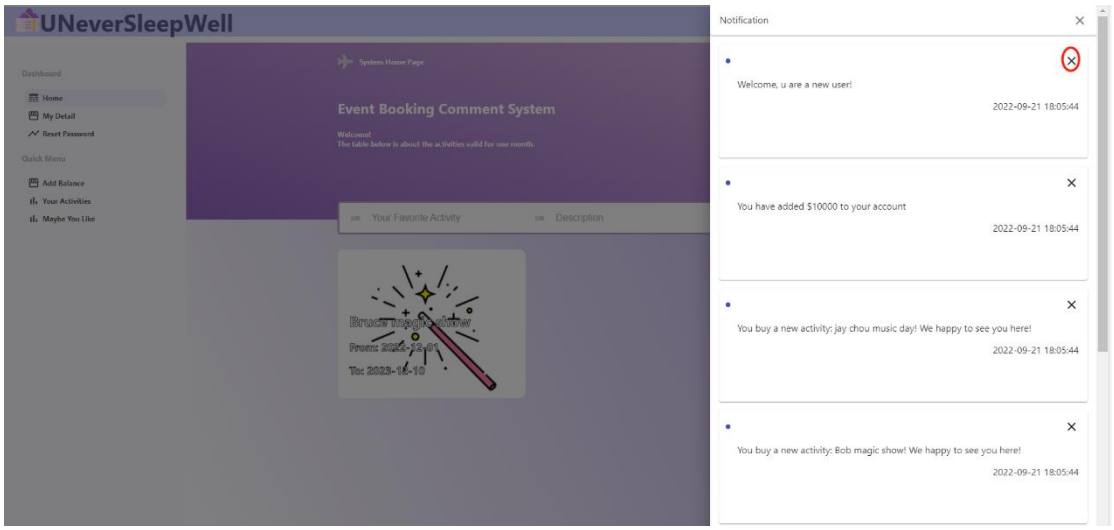
User all notifications – [Fundamental functionalities 6]
By clicking all notifications, the user should be online, and all notifications of the user would be displayed with unread notifications be marked.





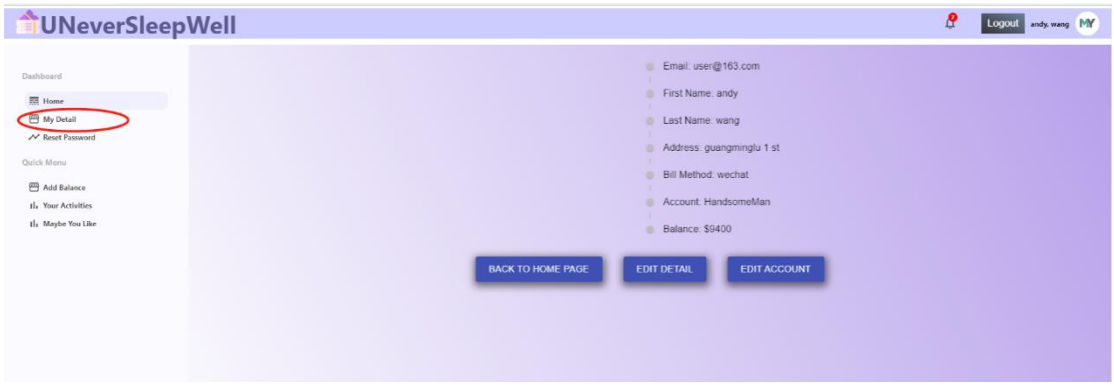
User delete notification – Fundamental functionalities 6

By clicking delete notification, the user should be online, and the notification would be removed from all notifications and database.



User detail – Novel functionalities 2

By clicking my detail, the user should be online, and they can check all their detail including email, first name, last name, address, bill method, account, balance.

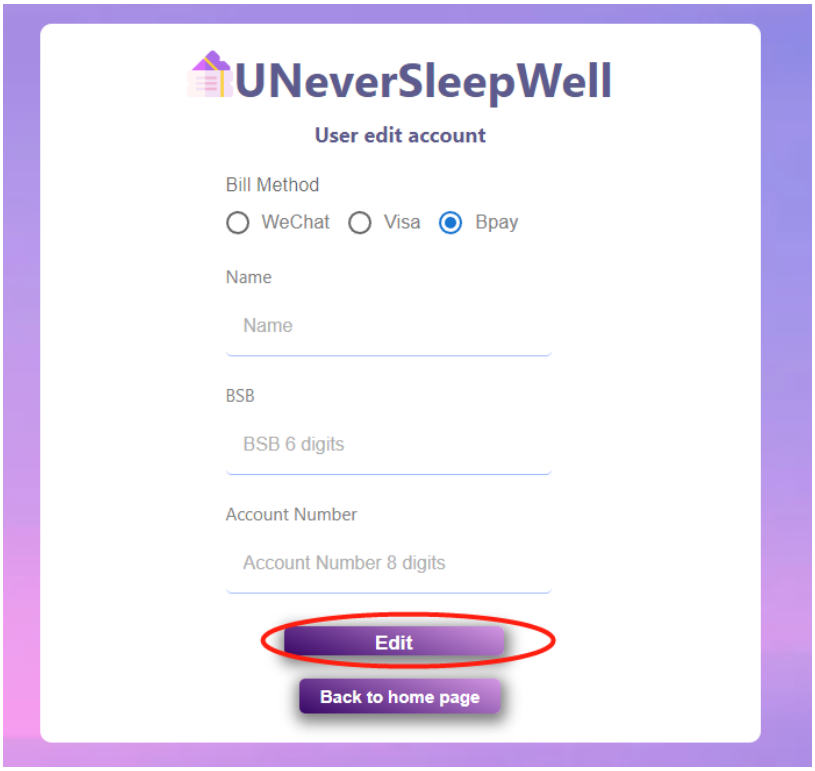


User detail update—Novel functionalities 2

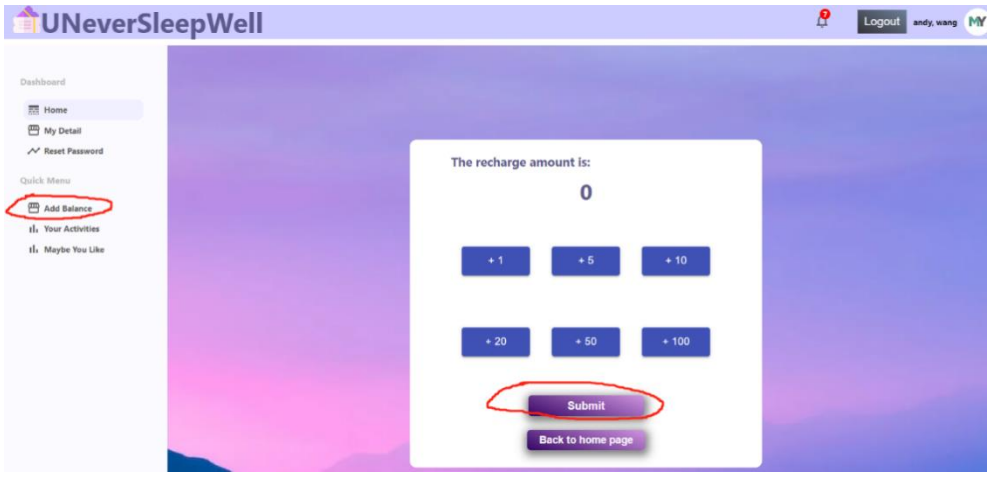
By clicking cancel activity, the user should be online, and the date should be 7 days before the activity begins. If both satisfy, then the user cancels this activity and gets money back to their account.

User edit account – Novel functionalities 2

By clicking edit account, the user should be online, and they can choose another payment method to change. There are three different payment method, and they can choose any one and type correct information to change their payment method.



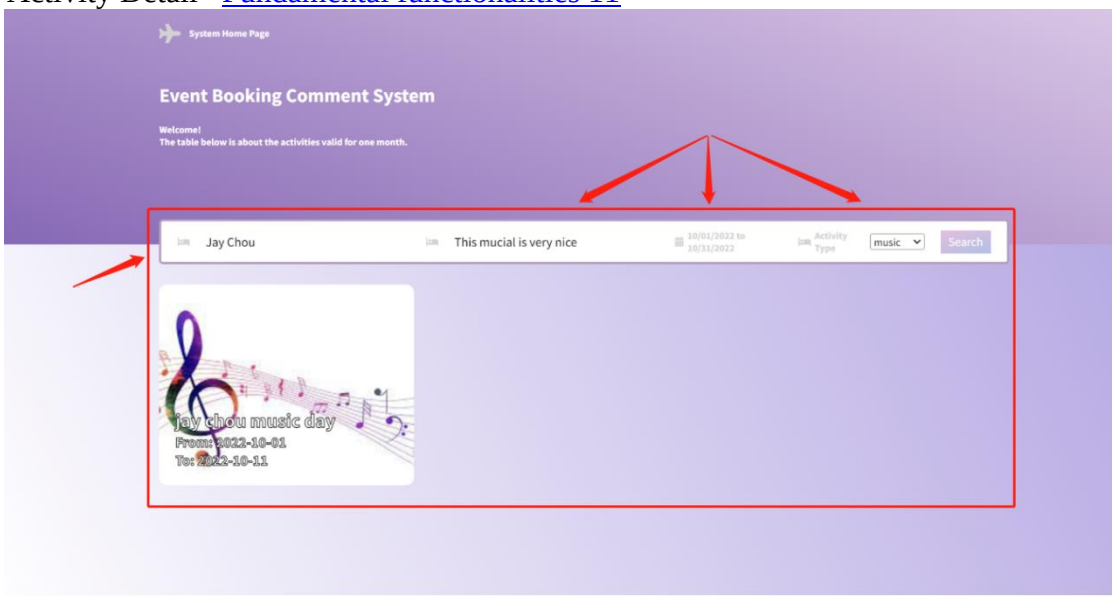
User add balance – Fundamental functionalities 5
By clicking add balance, the user should be online, and money should be greater than 0. If both satisfy, then money will be added to the user's account.



Search Activity –Fundamental functionalities 12

Host/Customers can search activities by entering activity part name/full name, description and/or type to find activities that fit all the customers' requirements. It will return a list of activities for customers, and if there is no matching, it means no activities.

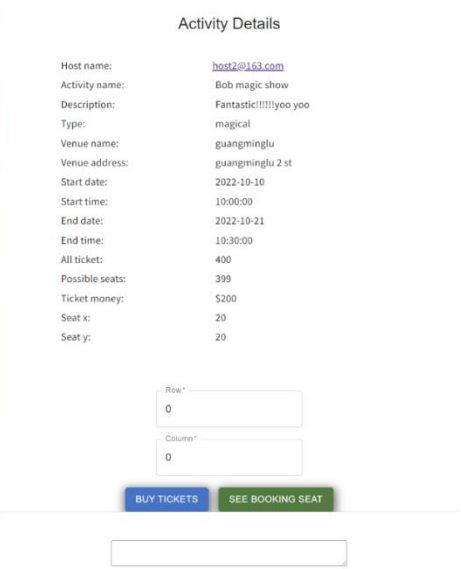
Activity Detail –Fundamental functionalities 11



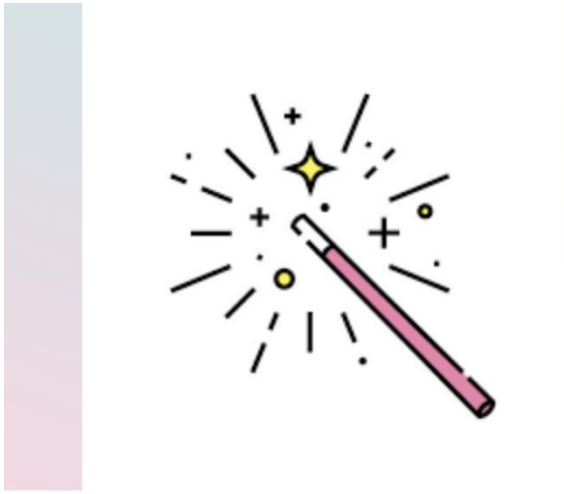
Customers can have a view of one activity with all the details with it: name, description, type, time, address, price etc… And the customers can also purchase the ticket at activity detail page. Picking seats, they like and also can leave a commit/review after the activity is over. Prospective customer can only check the detail and view all the commits, there is no purchase button for them.

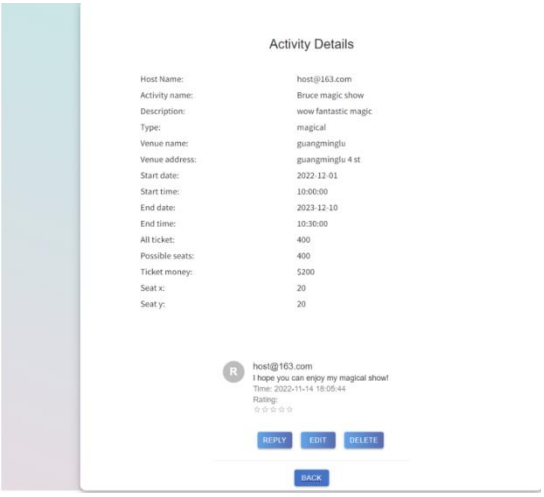
For host, there is only the commit button that is use for host to leave message to all the customers and prospective customers to review.

(For customer)

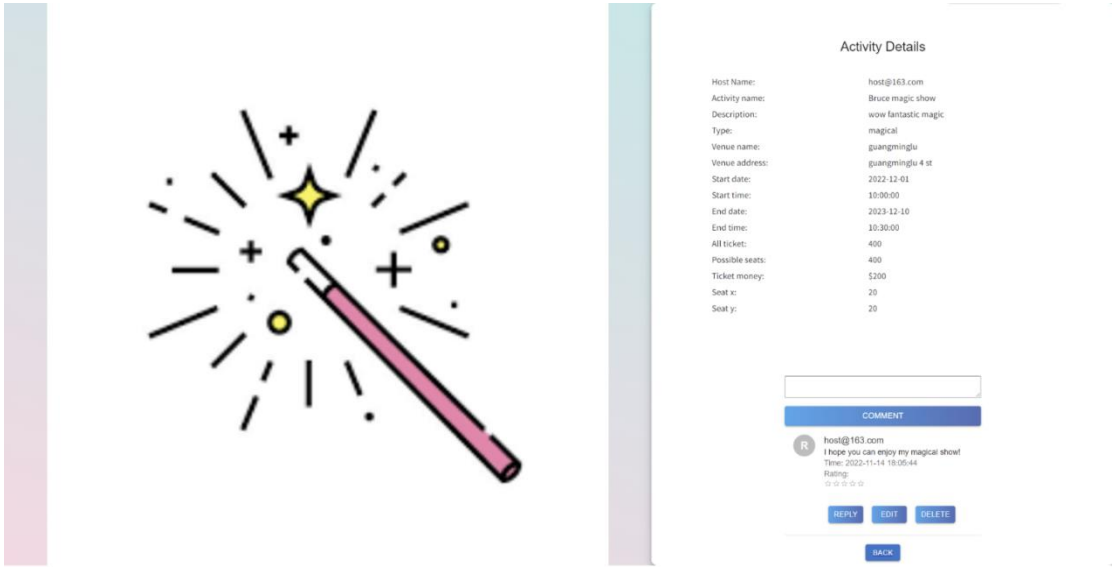


(For prospective customer)



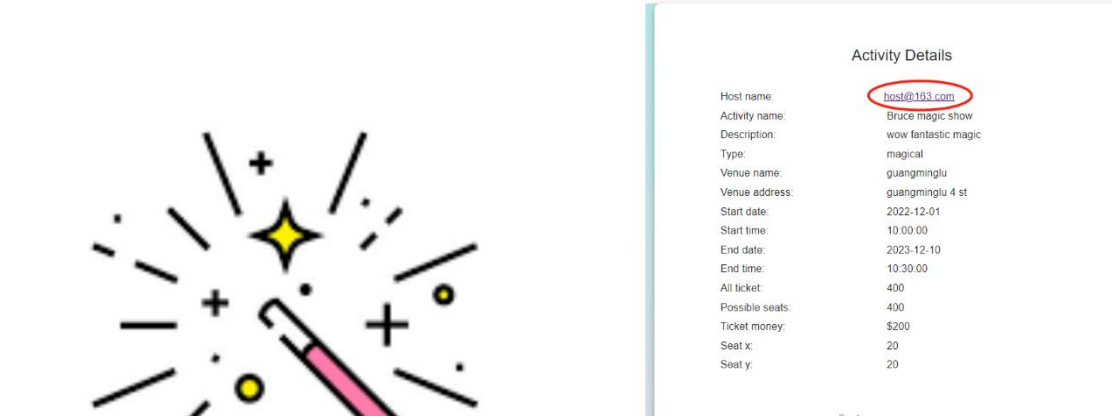


(For host)



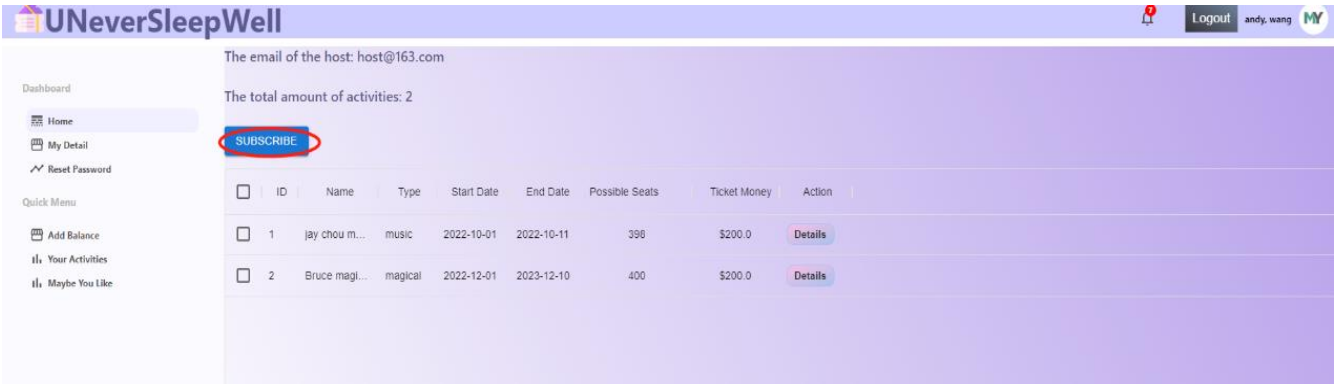
Host detail – [Novel functionalities 3]

By clicking the host's email address with underline, the user should be online and they can check the host's information and all activities booked from the host.



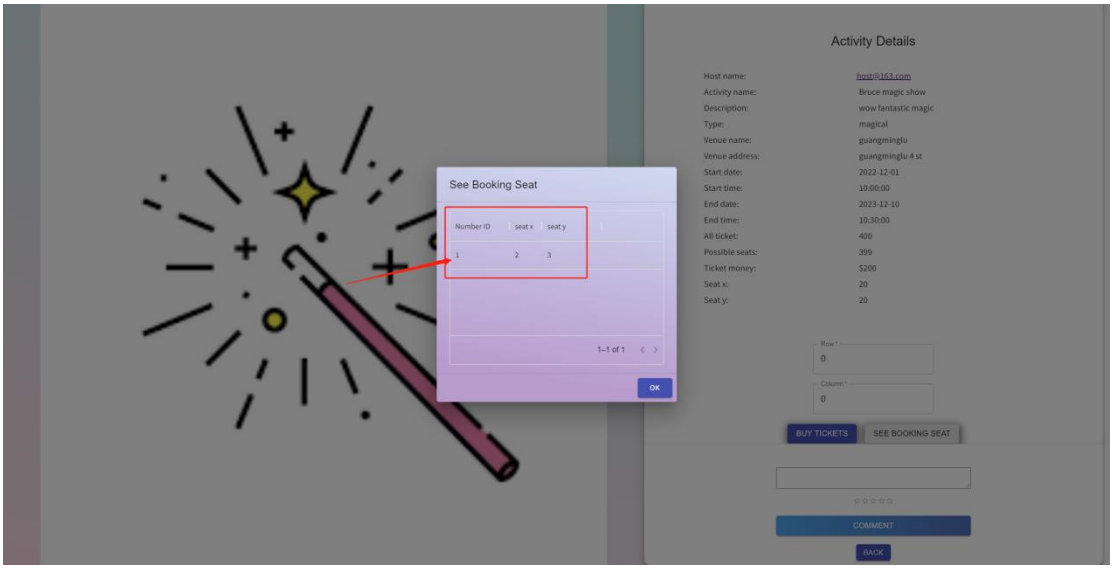
User subscribe host – Novel functionalities 3

By clicking the subscribe button, the user should be online, and they can choose to subscribe or unsubscribe the host. User will get notifications if their subscribed host creates new activities.

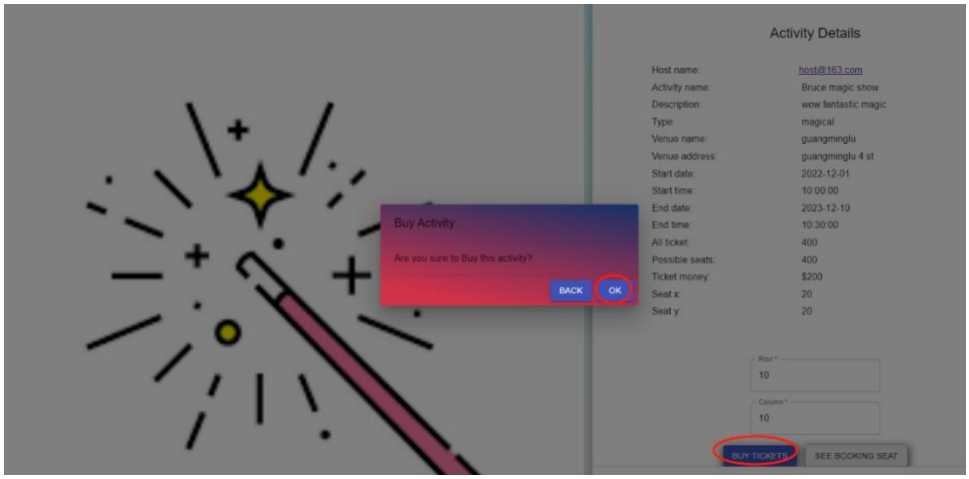


See booking Seats –Fundamental functionalities 5

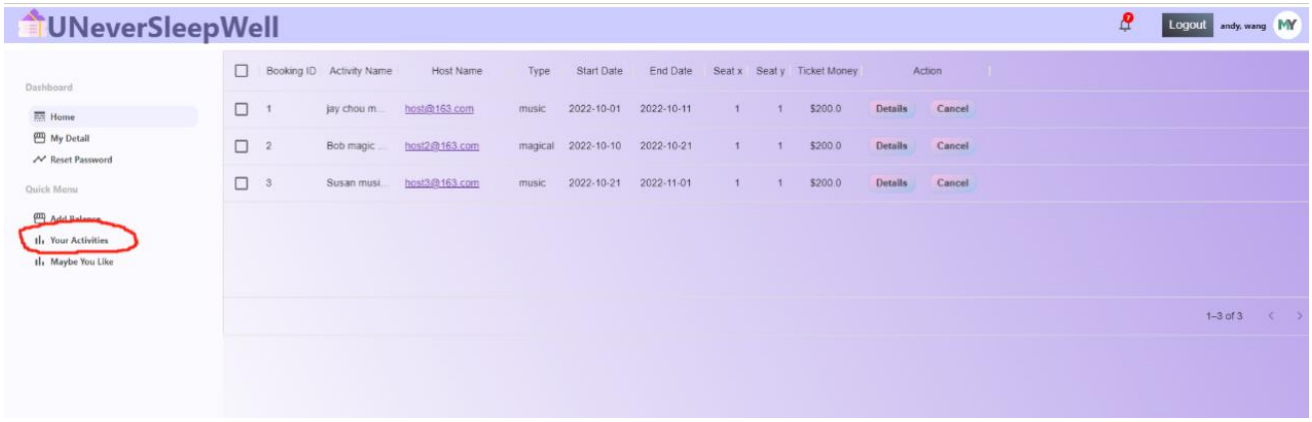
As the customer is booking the ticket, he/she must pick a seat, customer can click on the booking seats to see which seats are sold out already.



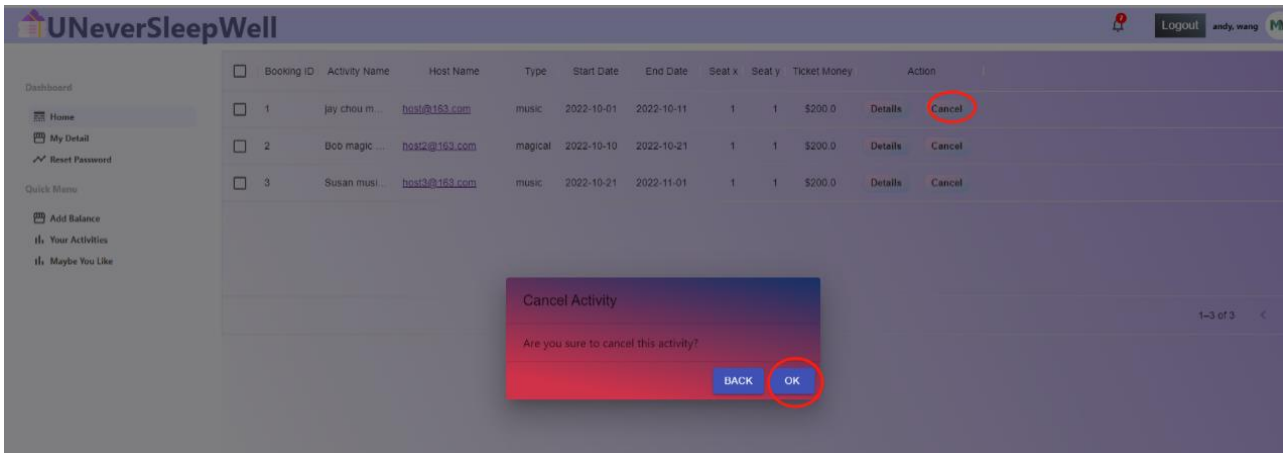
User book activity – [Fundamental functionalities 5]
When booking activity, the user should first choose an activity and choose a seat. If the money in their account is enough, then the activity will be booked successfully. If not, then they need to add balance and then book the activity again. If the seat is full, then they need to choose another seat to book, and after book successfully, a notification would be sent to the user.



User list activities – [Fundamental functionalities 5]
By clicking Your Activities, the user should be online and all the activities they book before will be listed here. They can choose to check activities detail or cancel their activities.

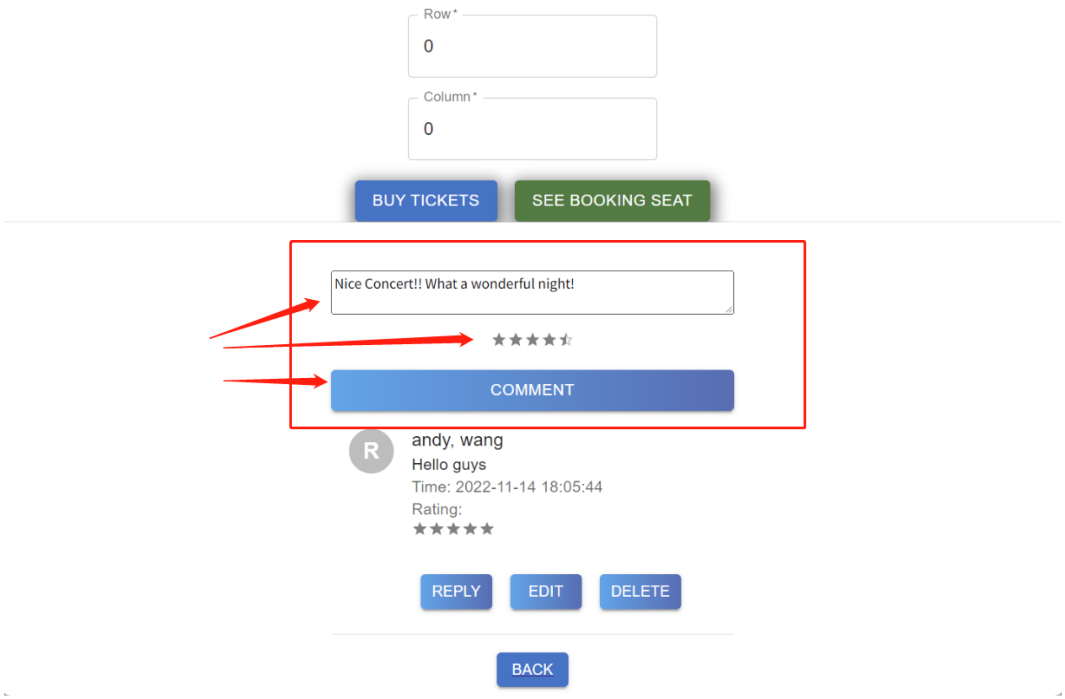


User cancel activity – Fundamental functionalities 8
By clicking cancel activity, the user should be online, and the date should be 7 days before the activity begins. If both satisfy, then the user cancels this activity and gets money back to their account.



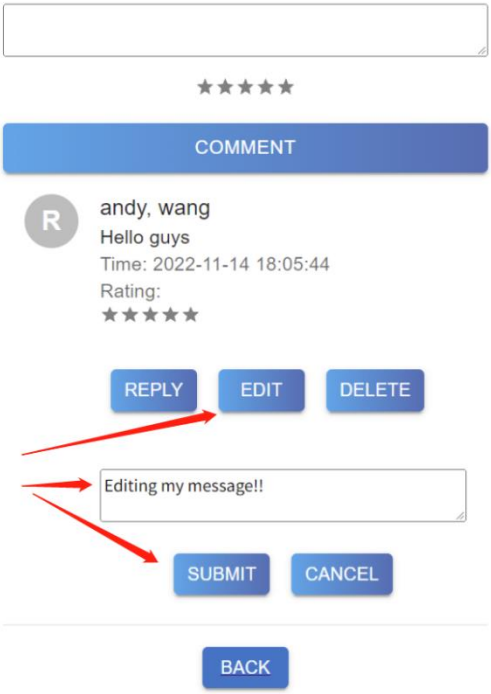
Add Commits/Reviews –Fundamental functionalities 9
After customer attend that activity, that customer can leave a commit/review of that activity. But it must wait until when the activity is over. And only one commit/review for each customer who have attended that activity. Customer can give rating of this activity of scale 0-10. The overall rating of the activity will be based on the average of all customer's rating.

Beyond that, Host can also commit his activity. And there is no one commit/review limit for user. But Host cannot give rating to his activity.



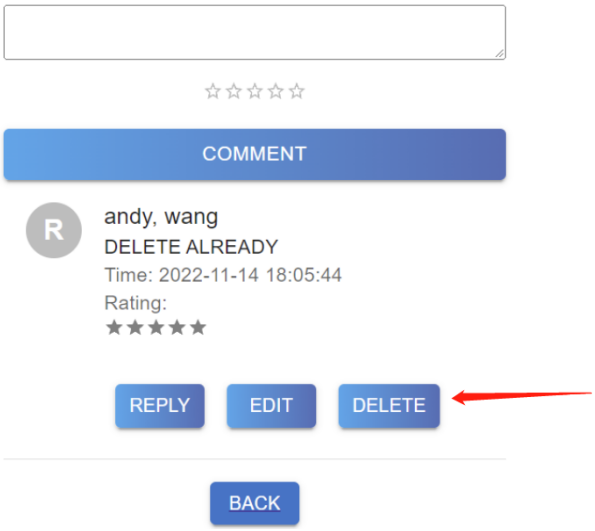
Edit Commit/Review – Novel functionalities 4

If the customer wants to edit of change the commit, he/she already send out, our system has edit function. The customer can add more new things, delete some of words and re-rating in the commit/review of his, and republish it with date. The host can also edit his commit/review of his own.



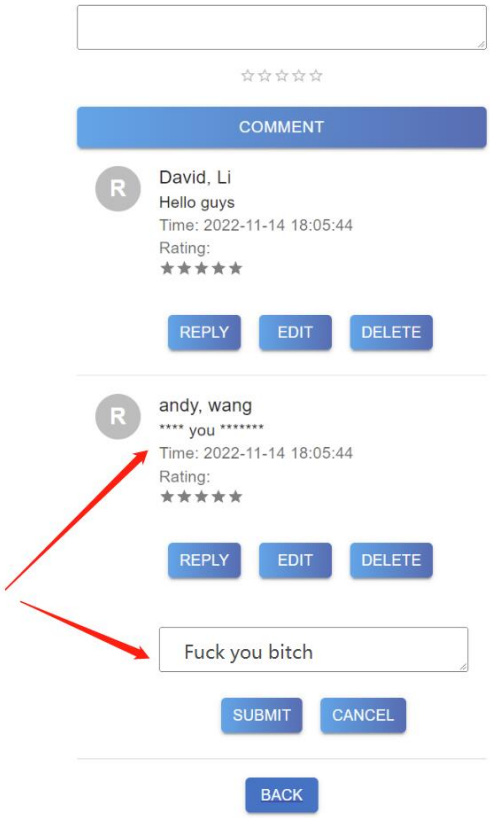
Remove Commit/Review – Novel functionalities 4

If the customer/host want to delete the commit/reviews or reply, they have already published. Our system provide delete system. It will delete the message that customer/host posted before. And the message will replace by DELETE ALREADY.



Commit/Review Filter – Novel functionalities 5

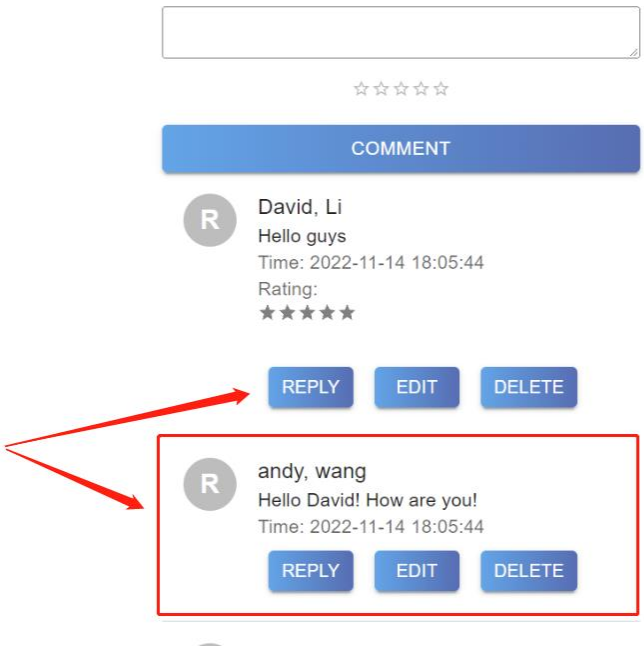
When a customer of a host wants to add a commit/review to the activity. Our system provides a language filter algorithm (DFA). It means that no bad word can be display in the commit/review area. So if host or customer want to leave a mean commits/reviews. It will automatically find the bad word and change it to ****.



Reply Commit/Review – [Fundamental functionalities 10]

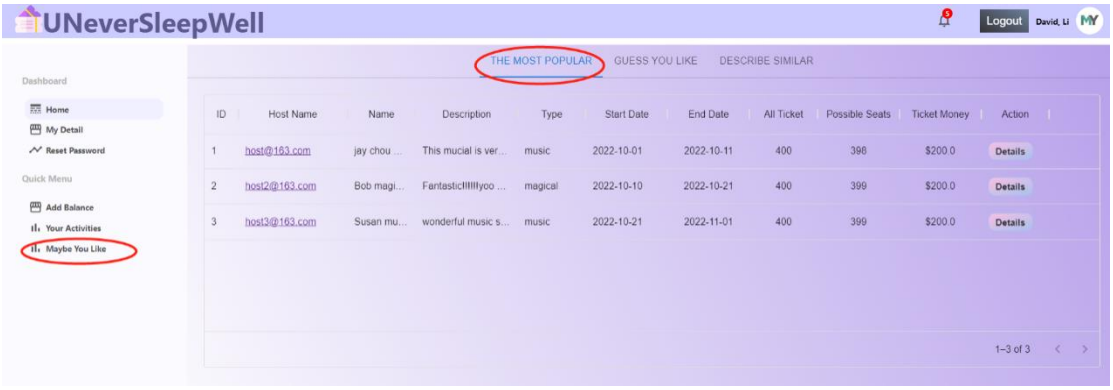
For host, he can reply to any commit/review that have been left in his activities.

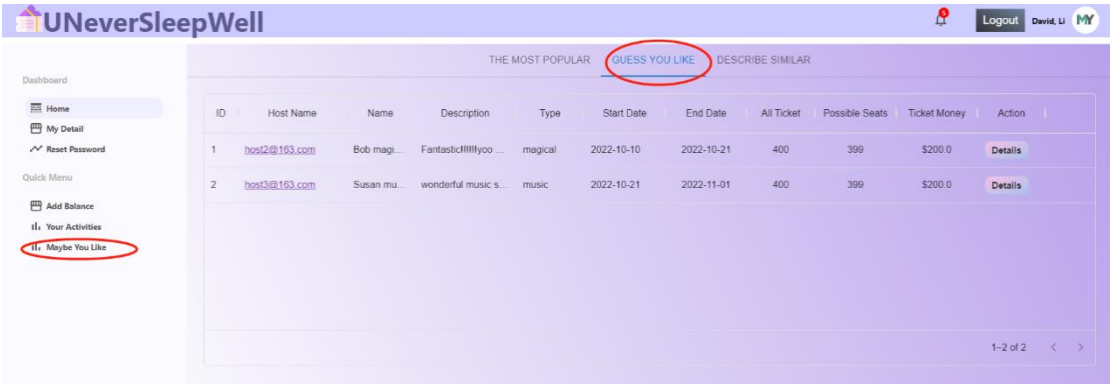
Customer can also reply to comments left by other customers and discuss with them. Commit is one for per customers, but reply does not have limit.

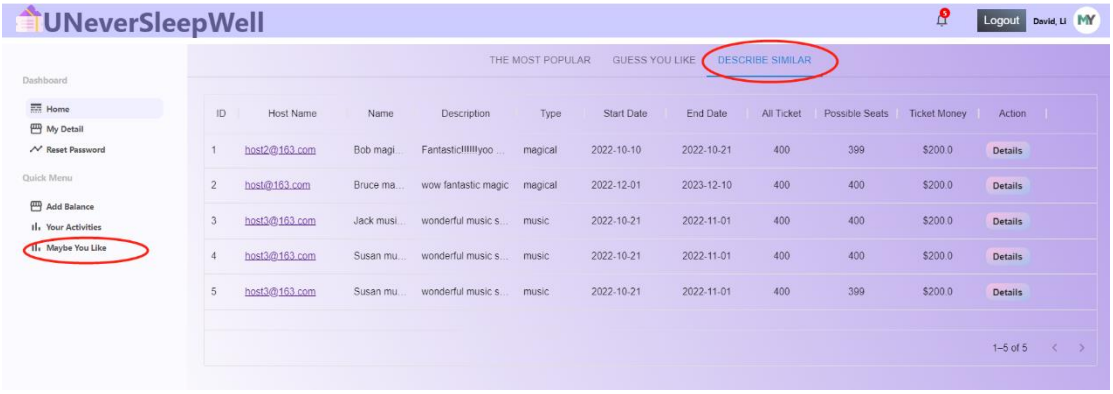


User Recommendations – [Fundamental functionalities 13]

There are three distinct recommendation systems available to users after they sign up. We offer the most popular and most sold activities right now available for new customers. For customers who have made purchases matching a given profile, we may recommend further purchases matching similar or nearly equal criteria. We may push additional personalized activities to him based on the similarities between users for those who have previously rated certain activities.







Implementation Challenges

Database:

Threading problems

First, while responding to the front-end with many requests, our server will present some challenges. Due to concurrent information queries, many instructions replying at once might confuse the MySQL database. We struggled with this issue for a considerable amount of time and tried to find a solution repeatedly without any success. Finally, in week 10, we found a way to change the Mysql query process to use thread locking to a single process, allowing only one query process to implement it. Prior to this, we tried using a thread lock on Flash's backend response or making the server more highly concurrent, but none of these methods were successful.

Transformation of strings and databases

There are issues with string splicing in database queries. The technique he employs can't combine numerous strings because of the fixed security database query. However, it is extremely effective to extract the identical components and group them into helper methods in order to increase the percentage of code that is reused, but we also want to prevent SQL injection at the same time. Therefore, to make the characters used into a queryable string, we first anti-inject the characters used and then stitch them together. This makes the database code more reusable and permits querying while maintaining database security.

Binary difficulties with databases

To improve the site's usability by adding graphics to the campaign. We have tried to upload local photos to the database. But because the database saves the binary by utilizing unique ways for storage and retrieval, this is quite challenging. Not to mention, storing it as a JSON file negates the purpose of having a database in the first place. In order for the front-end to display the images and the back-end to avoid having to perform additional operations on the images, we resorted to extracting the images directly from the website and merely storing the website address.

Back end

Format both the front and back ends

The data format is a crucial issue on both the front and back ends. The differing names and forms of the data in the front-end and back-end responses presented us with numerous challenges before we finally finished the system. The front-end found it challenging to partition the back-data end's structure. Additionally, on occasion, the front-end found it extremely challenging to function further due to the names' resemblance.

DFA

The challenge of writing a DFA algorithm is that you have to understand how DFA works clearly. DFA algorithm uses the principle of the map over the map to greatly optimize the retrieval performance with a time complexity of less than or equal to O(n). Moreover, when writing this algorithm, ensure that the performance is no longer limited by the number of keywords but only related to the length of the retrieved text.

Recommendation System

Popularity-based, user-based, and description-based recommendation systems are the three types that our system offers. Since we do not have much experience in developing user-based user-based recommendation systems in UNSW. Therefore, to complete this system, we have done a lot of research on how to develop user-based recommendations, calculate the similarity between users, and other related algorithms. Additionally, we looked at how to do rigorous text data preprocessing, convert text data into vectors or matrices and calculate the similarity between these vectors. And finally, by sorting the similarity, infer the activities that the user may like.

SMTP

When forgetting a password, we choose to use SMTP to send a random validation code to the host/user’s email to let them use the validation code to reset their password.

We use the smtplib module to connect to the email server, log in to the email, and send emails (with sender, recipient, and email content).

Use an email module to construct the email page, including sender, recipient, subject, body, etc.

STMP is quite useful to do such functionality, so we spend time on study and try it, and finally present a beautiful functionality to allow the host/user to get back their account when forgetting their password.

Front end

Cross-domain problem

In the scenario where front-end and back-end are separated, the most intuitive reason for encountering cross-domain problems is that the front-end and back-end are deployed separately on two machines or use different port numbers. When the front-end accesses the back-end service, no data or did not have the desired effect. The essence is that resources under one domain (including documents or script programs, etc.) try to access resources under another domain. After conducting some research, we found Cross-Origin Resource Sharing (CORS) could be a suitable solution. The CORS mechanism supports secure cross-origin requests and data transfers between browsers and servers. Modern browsers use CORS in APIs such as XMLHttpRequest or Fetch to mitigate the risks of cross-origin HTTP requests. In the end, our completely separated front-end and back-end projects got stable links.

Dynamics

Dynamics is a whole new concept for the front end. The majority of folks don't have a foundation in front-end because of how the course is currently designed in NSW. Therefore, front-end was a really challenging task for our crew. Near the end of the project, online searches for dynamic animations were made. We tried to take a variety of approaches. In the end, we settled on dynamic animations that would complement other elements while not detracting from the overall impression. For this, we made the decision to use falling snow and confetti. The main page's snowfall can move in accordance with the mouse's position. At the same time, we gave each button a three-dimensional appearance by applying gradient colors to give them a sense of depth. To make the buttons appear in 3D, we employed shadows.

Visualization of charts

There were numerous occasions when we tried to visualize charts using the charting module, but the page would just show a white screen or a jumble of jumbled codes. E-chart lessons are widely available online, but there aren't as many resources can use for reacting projects or for our project to employ. We put a lot of time into finding resources that were available and determining whether they might be applied to our

project. We found it really challenging to transfer the data into a graphical data format. Additionally, the database was empty at the start of the project, thus all data started out at zero. The charts' limitations in this situation would result in a white screen on the page. Therefore, we implemented a filtering process in the backend to ensure that the zero and empty charts are properly displayed when the project data is empty.

Encryption

Although utilizing JWT to transport tokens between the front and back ends of a communication process is an efficient method, we discovered while considering this method that it is ineffective at thwarting replay assaults. As a result, we pass both ID and token while using a separate setting to guarantee the encryption's validity. Then, because the token is generated at random and the previous token could not match the existing id, when the hacker obtains the id or token, it will result in a situation where the id and token do not match. This ensures the token's validity, which means that it won't be valid for a long time and will expire quickly, and that the user will receive a fresh token each time he logs in.

Descriptions of third-party functionalities

React
React is a declarative, effective, and adaptable JavaScript user interface library. It enables you to build intricate user interfaces out of discrete, little chunks of code known as “components”. We choose to react as our front-end framework because of the following reasons. React makes JavaScript coding much easier, handles dependencies and has an excellent cross-platform support.

Material UI
Material UI or MUI is one of the best UI frameworks for ReactJS that integrates React with Google’s material design. We choose Material UI for the reason that it provides great help in creating designs with fully functioning code components. In addition, Material UI is highly customizable so that we can add more styles while keeping the functionality working.

Flask
Flask is a web framework written in Python, which allows us to quickly implement a website or web service using the Python language. We chose Flask to link our backend functions with our frontend. We used lots of POST, GET, and DELETE methods to transfer our data between frontend and backend.
The performance of flask meets the needs of general web development, and its flexibility and scalability are superior to other web frameworks.

SMTP
SMTP (Simple Mail Transfer Protocol) is the Simple Mail Transfer Protocol, which is a set of rules used to transmit mail from the source address to the destination address, and it controls the transfer mode of the letter.
To send emails, we need two modules, smtplib and email. The smtplib module is mainly responsible for sending emails, and the email module is mainly responsible for constructing emails. These two modules come with python, only need to import them to use them.

PyMySQL
MySQL serves as the database management for this project. Python can interface with SQL databases thanks to this module, enabling us to use Python to run SQL code.

MySQL is a fairly developed database that allows for a lot of complicated operations, thus that is why we chose this module. It also includes built-in directives for security and to avoid SQL injection. The results of the entire project are unaffected by this module.

Cypress

In order to make sure that our project functions properly on the front end, we also employ front-end testing. Cypress makes it possible to test any browser-based content quickly, easily, and reliably. As a library to view the status of tests running on the front end, this library is more user-friendly for developers. We can observe what activities are currently needed on the front-end in accordance with our instructions as we build and run this test. This library is free and has no impact on our project. (JavaScript End to End Testing Framework | Cypress.Io Testing Tools, n.d.)

jiaminghi/data-view-react

The guide at http://datav-react.jiaminghi.com/. We searched the web for this website to get some advanced charts that would help us visualize all the data. It has quick and effective data visualization capabilities. It is an open-source, free library. As a result, we used this library to plot our summary data, which is accessible on the homepage. No effect is felt on our project.(DataV-React, n.d.)

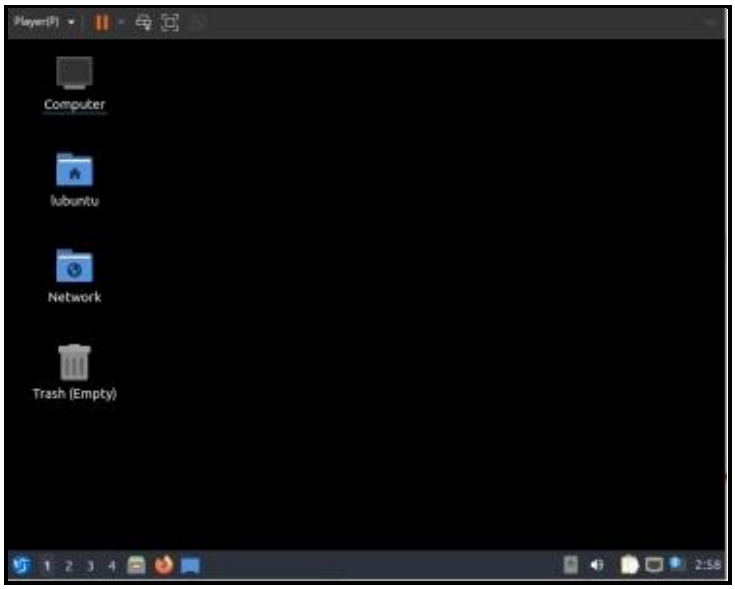
NLTK

NLTK (Natural Language Toolkit) is a toolkit built for working with NLP task in Python. It provides various text processing algorithms such as tokenizing, stemming, sentiment analysis, topic segmentation, etc. Our activity recommendation algorithm actually needs to compare the textual similarity of the descriptions of the activities the user participated in to the descriptions of other activities they did not participate in. Therefore, using NLTK for text preprocessing of activity descriptions can bring great convenience for the further comparison or analysis. (NLTK :: Natural Language Toolkit, n.d.)

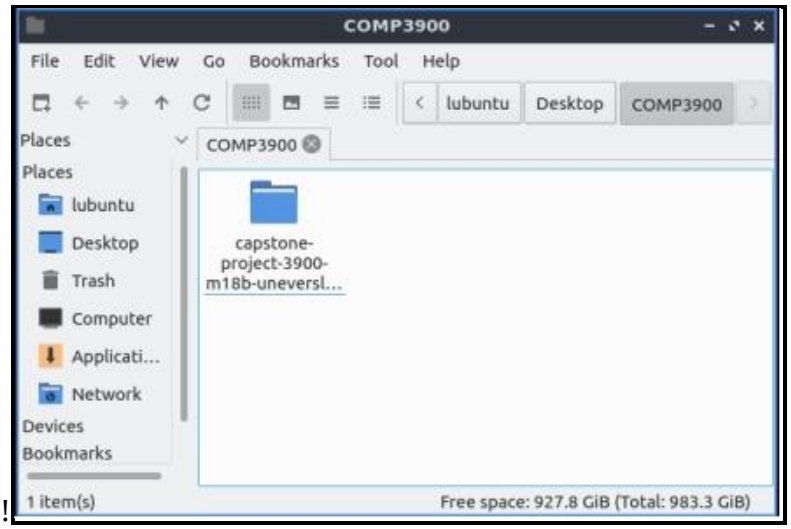
User Document/Manual

VMware Workstation 16 Player

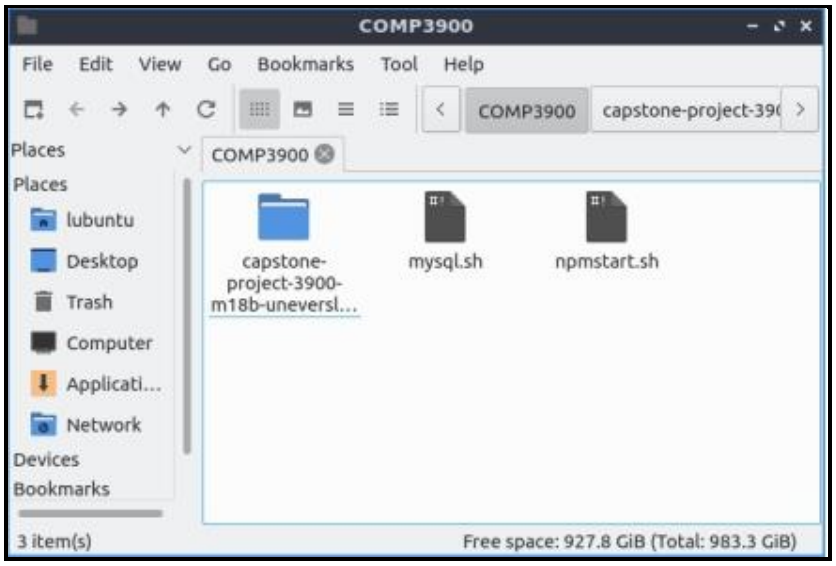
Set up VMware until you get to this page



Go to https://github.com/unsw-cse-comp3900-9900-22T3/capstone-project-3900-m18b-uneversleepwell and download the zip file (Use the account that has authority! Make sure you are a member of this project). Save it to Desktop, and also create a new folder called COMP3900. Then extract the zip file to the folder COMP 3900. After all this, the folder should look like this:

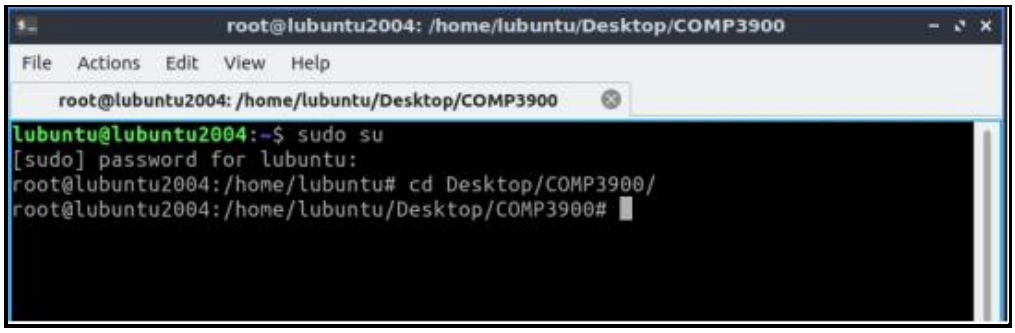
! 

Click into capstone project-3900-m18b-uneversleewell, and select mysql.sh and npmstart.sh these two files, copy and paste them to the upper level directory (below):

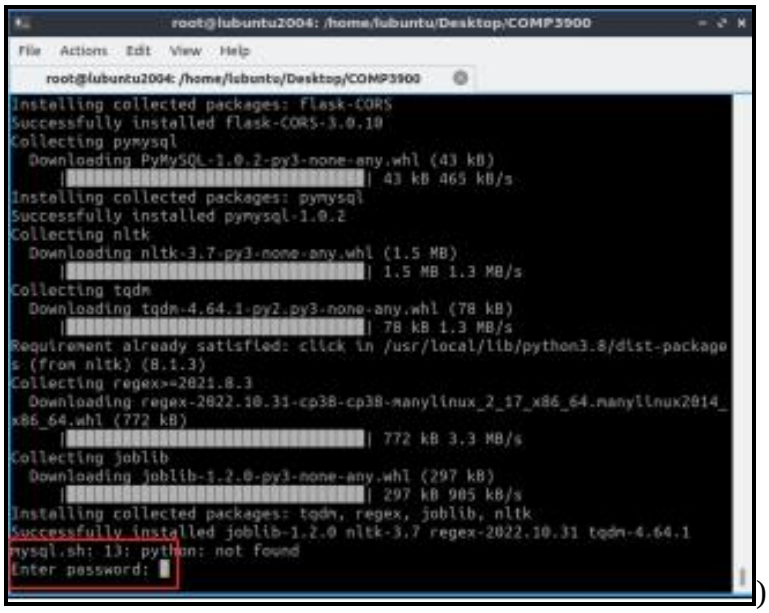


Then open the terminal, enter sudo su and the password (lubuntu).

Next, go to folder COMP3900:



Run sh mysql.sh first to install all the requirements for MySQL. (In the installation, you need to enter ‘Y’ manual) After this script finished, it should show like this:



Enter any random password you want at this moment, and you will enter MySQL successfully. Therefore, enter the following commands to change the root password of MySQL:

```
select user, plugin from mysql.user;

ALTER USER ‘root’@’localhost’ IDENTIFIED WITH mysql_native_password by ‘0000’;

flush privileges;

exit;
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement
.

mysql> select user, plugin from mysql.user;
+------------------+-----------------------+
| user             | plugin                |
+------------------+-----------------------+
| debian-sys-maint | caching_sha2_password |
| mysql.infoschema | caching_sha2_password |
| mysql.session    | caching_sha2_password |
| mysql.sys        | caching_sha2_password |
| root             | auth_socket           |
+------------------+-----------------------+
5 rows in set (0.00 sec)

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by
 '0000';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> exit;
Bye
root@lubuntu2004:/home/lubuntu/Desktop/COMP3900#
```

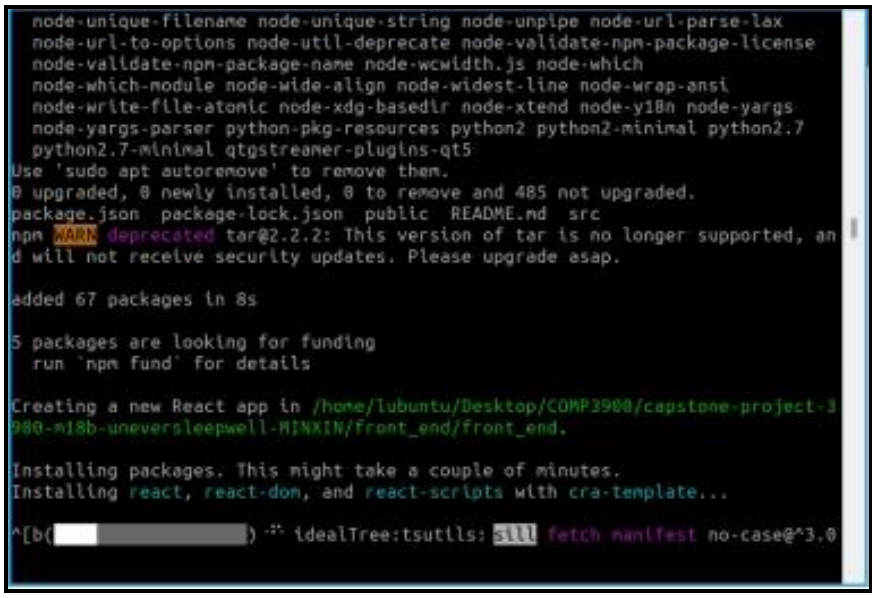
At this point, you are good to go with MySQL. The next thing to is to install React environment, run `sh npmstart.sh` to complete React installation.

When you run `sh npmstart.sh`, it should return error or get stuck at some point, don't worry, just run `sh npmstart.sh` again, the problem will be solved (Error showed above). The problem might be network problem.

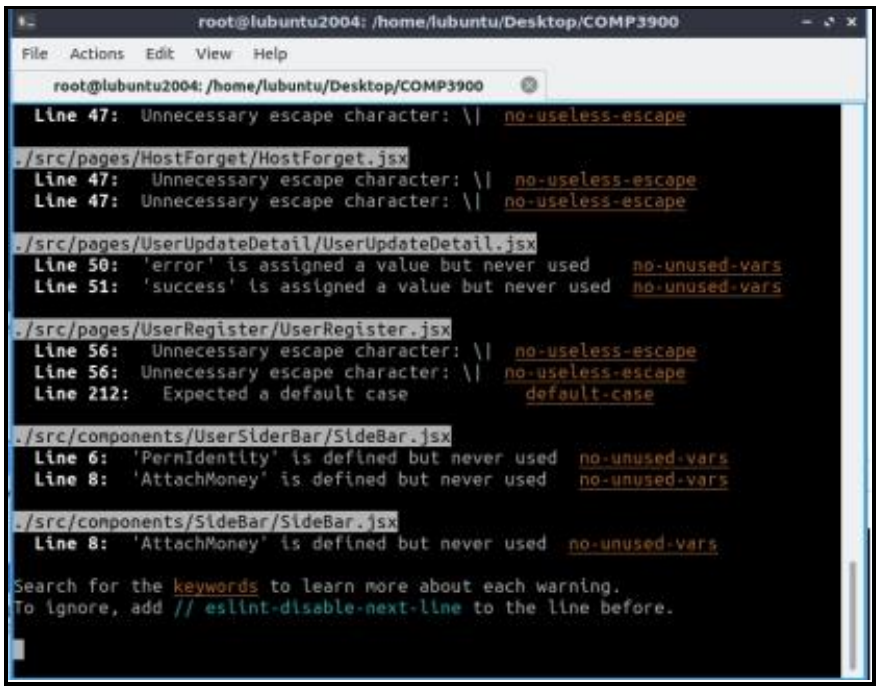
(Error you might see)

```
/home/lubuntu/Desktop/COMP3900/capstone-project-3900-m18b-uneversleepwell-MIN
XIN/front_end
ERROR: npm v9.1.1 is known not to run on Node.js v10.19.0. You'll need to
upgrade to a newer Node.js version in order to use this version of npm. This
version of npm supports the following node versions: `^14.17.0 || ^16.13.0 ||
>=18.0.0`. You can find the latest version at https://nodejs.org/.

ERROR:
/usr/local/lib/node_modules/npm/lib/utils/exit-handler.js:22
  const hasLoadedNpm = npm?.config.loaded
                         ^

SyntaxError: Unexpected token .
    at Module._compile (internal/modules/cjs/loader.js:723:23)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:789:10)
    at Module.load (internal/modules/cjs/loader.js:653:32)
    at tryModuleLoad (internal/modules/cjs/loader.js:593:12)
    at Function.Module._load (internal/modules/cjs/loader.js:585:3)
    at Module.require (internal/modules/cjs/loader.js:692:17)
    at require (internal/modules/cjs/helpers.js:25:18)
    at module.exports (/usr/local/lib/node_modules/npm/lib/cli.js:76:23)
    at Object.<anonymous> (/usr/local/lib/node_modules/npm/bin/npm-cli.js:2:2
5)
    at Module._compile (internal/modules/cjs/loader.js:778:30)
```

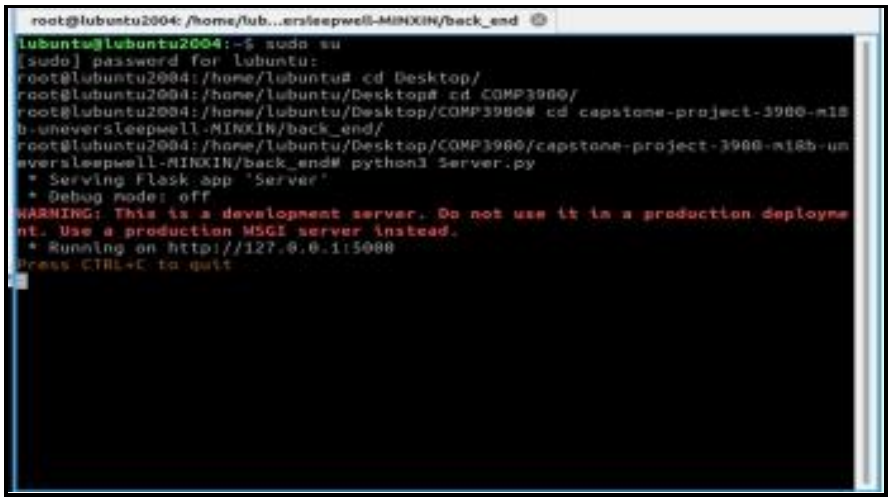
(Here you might get stuck)



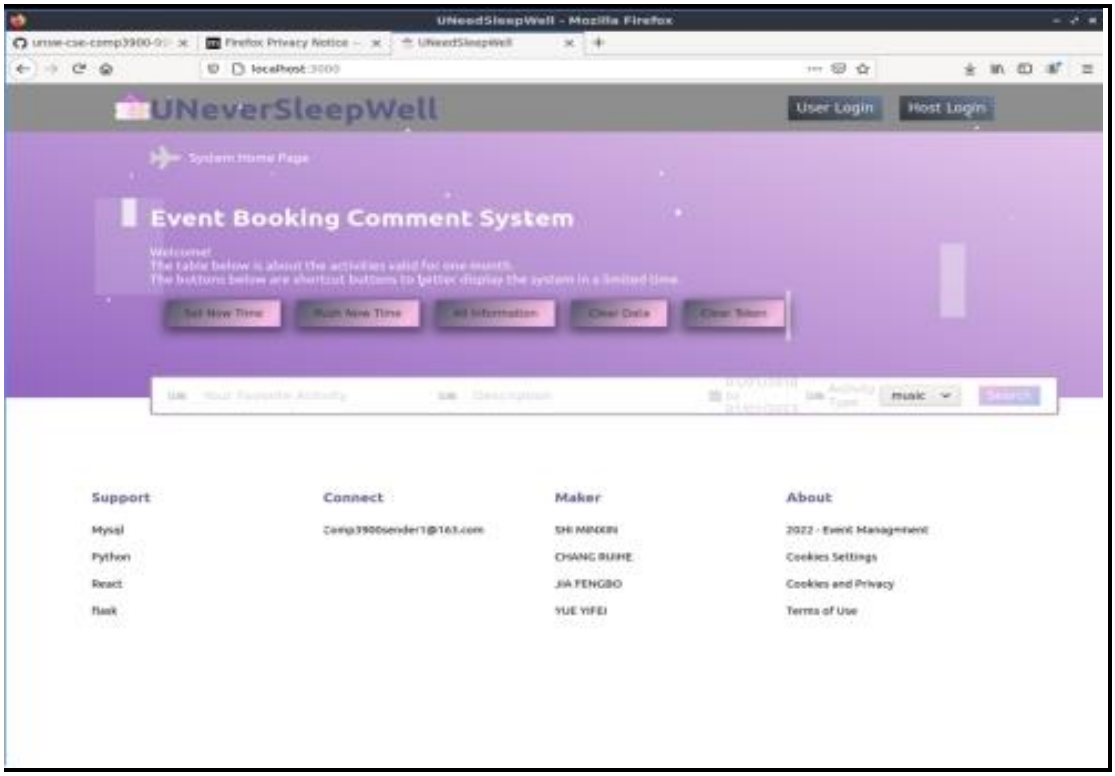
When you see the diagram below, it means React was installed successfully.



The next thing you need to do now is open another terminal and go to the back-end file to start Server:



And then you can go to fire fox and type “localhost:3000” to see the website finally.



Windows

Not the perfect tutorial, the perfect tutorial is the virtual machine version. Here just some simple steps on window (may be mac also can using this guide).

https://www.mysql.com/cn/downloads/

Download a stable version of MySQL, set passport as 0000(or you can change it in begin.py as you like)

https://nodejs.org/zh-cn/download/

Download a stable version of node (best 16)

Use the following instructions in our project directory:

```
sh windowsh.sh
```

```
npx create-react-app filename(you can choose)
```

```
cd filename(you can choose)
```

move the files in front end into this file(filename)

```
npm install
```

```
npm i @mui/lab @mui/material
```

```
cd back_end
```

```
python3 back_end/Server.py
```

```
cd ..
```

```
cd filename
```

```
npm start
```

It is worth noting that our system has both front-end tests and back-end tests. The coverage report of our back-end tests can be run according to the following instructions. First, we go to the back-end folder. Then run those instructions.

```
coverage run -m pytest
```

coverage html

Wrote HTML report to htmlcov\index.html

For the front end, please go to the front-end folder and run those two instructions on different terminals

npm start

npm run sypress

Then select the first E2E test according to the pop-up page. Then choose a browser to run the test, you can see that we have several test folders. Run these test folders to see our front-end tests.

References

Architecture of Business Layer working with Entity Framework – The Reformed Programmer. (n.d.). Retrieved September 29, 2022, from https://www.thereformedprogrammer.net/architecture-of-business-layer-working-with-entity-framework/

Database Layer. (n.d.). Retrieved September 29, 2022, from https://cis.msjc.edu/IA/DatabaseLayer/

DataV-React. (n.d.). Retrieved November 18, 2022, from http://datav-react.jiaminghi.com/

JavaScript End to End Testing Framework | cypress.io testing tools. (n.d.). Retrieved November 18, 2022, from https://www.cypress.io/

Li, S. (2017). Security Requirements in IoT Architecture. *Securing the Internet of Things*, 97–108. https://doi.org/10.1016/B978-0-12-804458-2.00005-6

NLTK :: Natural Language Toolkit. (n.d.). Retrieved November 18, 2022, from https://www.nltk.org/

What is Three-Tier Architecture - Singapore | IBM. (n.d.). Retrieved September 30, 2022, from https://www.ibm.com/sg-en/cloud/learn/three-tier-architecture