

# Report

## Website:

<https://wander-land.herokuapp.com>

## Description:

Wander-land is a platform made for traveling enthusiasts and for those who want to show their national hospitality. Wander-land has two main functionalities: Looking for travelling partners and to provide tour guidance. Wander-land is an international based project. That means users can be from around the world no matter where they come from. An economical system for tour guides is also implemented in Wander-land for those who want to be a tour guide and earn extra cash at the same time.

## How it works:

Wander-land is a hybrid of social network and an online service for tour guide postings where people can interact to do business with. A login system is implemented for users to identify who they are in Wander-land. For those who have registered as users whose intention to find/join anyone to travel to the same destination, they can either “write” a post or search people or groups using the search bar to find people who match what they are looking. As for users who are interested in hospitality, they can post an advertisement for people to search them and potentially hire them as their tour guides.

## Implementation Details:

In Wander-land, we have used the MVC design pattern as our architecture. We have set up the database using PostgreSQL as the model. Whenever the user requested anything from our website, it goes to app.js, then to our controller file router.js from the folder controller. Then according to what was requested, our controller would fetch the data from our database that is related to what user was requested then send back to the controller and process it. After the some processing, it would send back the processed information back to the user on the font end.

For security, https was used in Wander-land for the protocol. We have also applied the hashing function sha-256 to our passwords for secure storage. As for the database, we have used a view instead of a table to store admin's information to prevent potential attack from using administration accounts to alter the database. We also used both blacking listing and white listing to filter any potential SQL injections. CSRF tokens were used in the app to prevent Cross-Site Request Forgery. On the font end site, we used form validation using both JavaScript and html to make it more secure.

For performance, we used relational database instead of NoSQL to prevent the any performance issues as our database grows. We also compress the files to

make the size smaller, so that rendering time would significantly increase. Responsive design is also implemented to make our app user friend, therefore more accessible to the public.

For unit testing, we used mocha, chai, and supertest modules to test our features. Our testing features include database connection, testing every links. When we tested our database, we used anything where id 1.

The functionality of our searching bar is to search any posts in the database that matches what the user had input. We have implemented an algorithm for if no such result was found. If the user was searching someone to travel with and no result was found, a recommendation would appear to user to who all the users who are looking for travel partners in both origin and destination. If the user was searching a tour guide, then it would display any tour guide posts within the same country in the destination.

For messaging, we implemented a system using socket.io. This messaging system is intended for sharing experience and memories and therefore everything entered will be stored in memory instead of our database. Users can only access this page if they have logged in and they would be told to do so otherwise. Users do not necessarily need to use their original info from our database but using their email as the identifier is recommended. Users will be able to create rooms, where other users can join. Each user can only create one room for simplicity. For this phase, we do not expect a large number of users staying in the same room; therefore there is no limit for number of users. Messages sent in a room can only be seen by other members in this room, and sending messages without joining a room is not allowed. When the owner of the room leaves the room, disconnects from the server, or delete the room, the room will be removed. All the other members of this room will be automatically removed from the very room. For future plans, we believe it is beneficial to add the ability to transfer images and files, and to enable video chatting.

For creativity, we have added an interactive map for users to find information's for countries easier. What the user has to do is to find the country where he/she wants to know more about then click on the area of the country then it would direct the user to the corresponding country informational page. More over we also added responsive design. Further more we have added some of the most beautiful photo from all around the globe to the slide show on the home page to increase over all user experience. A feature of restarting the server was implanted in Wander-land. If the server crashed, it would automatically restore the entire website 10 times. We have also implemented a "room" function in messaging to let users to create a room and give a name whatever the user want. On the search bar we have implemented suggestion and auto complete when users are searching in the city section as well as the calendar when the user is choosing a date.

## **Challenges:**

In certain countries, tour guides might need to obtain a licence in order to make a profit. At the same time it is hard to prove that their qualification since the law for every country is difference. Users who are travellers can also be hard to verify their identifications due to the multinational nature of Wander-land.