OULU UNIVERSITY OF
APPLIED SCIENCES

Kushal Atreya

# DIGITALISATION OF HOUSING IN NEPAL

Full Stack JavaScript Application

# DIGITALISATION OF HOUSING IN NEPAL

Full Stack JavaScript Application

Kushal Atreya
Bachelor's Thesis
Spring 2020
Information Technology
Oulu University of Applied Sciences

# ABSTRACT

Oulu University of Applied Sciences
Information Technology

Author: Kushal Atreya
Title of the bachelor’s thesis: Digitalisation of Housing in Nepal
Supervisor: Anne Keskitalo
Term and year of completion: Spring 2020 Number of pages: 46 + 2 appendices

Yarshaa DHN (Digital Housing Nepal) is a web-based start-up project which allows users to search, sell, and reserve the apartment. The traditional rental methodology is still practised in Nepal, where landlord and tenant meet with a local means of advertisement, personal contacts, and or word of mouth. This practise quantifies the time, effort, and resources for both hands and is arduous to get the proper match. Yarshaa DHN solves the complication by renting, featuring the rentals for the landlords, and by providing a visual dash-board platform for the tenants who can search the right apartment with the search and sort parameters such as location, size and facilities, from the web-based application.

Data were collected from primary and secondary sources. A survey was performed in Kathmandu where 76 people were participating by filling the form with a few questions. Among them, 12 were potential landlords and 64 were potential tenants. Furthermore, research-based findings and information from the Internet and publication were utilized. The waterfall methodology was implemented to develop the application, which consists of analysis, design, development, and testing phases, respectively. Modern web development tools, such as HTML5, CSS, Bootstraps, JavaScript, React, React router, Redux, Express and MongoDB, were extensively used to make the prototype responsive, modern, and interactive.

A full-fledged modern web-based prototype was developed (keeping the outcome of survey and business perspective coherent) to demonstrate what could be the face of change.

Keywords:

Web Development, React, Responsive, Housing, MERN, Full stack JavaScript, MongoDB, Express.js, Node.js, Redux.

# PREFACE

I participated in BUCSBIN (an interactive 5-day workshop held in Nepal) as a student of OUAS. During the workshop, we were cranking and developing innovative business ideas which can thrive in the market of Nepal by solving the everyday problem. The part of ‘ solving the everyday problem and creating a business out of it’ left an imprint on me. The basis of this thesis stemmed from the brainstorming session of the workshop while comparing the everyday lifestyle in Finland and Nepal.

In sincerity, I could not have accomplished my current level of success without strong supporting hands. First of all, Asabeneh Yetayeh for selecting me as a Bootcamp student and guiding me to excel in programming. My immediate thesis supervisor Anne Keskitalo and language instructor Kaija Posio for continuous constructive feedback. Anuj Thapa and Suman Kharel for brightening the idea. The Yarshaa team for helping me in the field survey and my family for who and where I am today.

Oulu, 11.12.2019
Kushal Atreya

## CONTENTS

## TABLE OF CONTENTS

# VOCABULARY

| Abbreviations / Terms | Meaning |
|---|---|
| 2D | Two Dimension |
| 3D | Three Dimension |
| 3G | Third generation |
| BOOTSTRAP | Front-end framework for developing responsive web-application |
| BUCSBIN | Building University Capacity to Support Business Incubation in Nepal |
| CDN | Content Delivery Network |
| CMS | Content Management System |
| CSS | Cascading Style Sheet |
| CSS3 | Latest version of Cascading Style Sheet |
| DevTools | Developer Tools |
| DOM | Document Object Model |
| HTML | Hyper Text Mark-up Language |
| HTML5 | Fifth version of the HTML |
| MacOS | Macintosh Operating System |
| MVC | Model View Controller |
| IoT | Internet of Things |
| JS | Java Script |
| JSON | JavaScript Object Notation |
| navbar | Navigation Bar |
| NoSQL | Non-relational or non-SQL |
| NPM | Node Package Manager |
| OUAS | Oulu University of Applied Sciences |
| UI | User Interface |
| UX | User Experience |

# 1 INTRODUCTION

Digitalisation means the transformation of existing business ideas while taking full use of technological advancement. It allows the business to grow, find the potential market and open to many dimensions. Almost every industry and business are taking full use of technological advancement to get ahead in their game. From eating habits, reading, making friends, buying and selling things, travelling, booking hotels, every integral part of our life has changed in a matter of a few decades. From doing everything manually, to now using the Internet, over the phone, you can check-in, and out of the hotel without requiring to see anybody. This not only shows how digitalisation is changing the world but it also proves that people are ready to adapt and embed.

Nepal was the very first country in Asia to adopt the 3G mobile telecommunications technology [1], Nepal widely uses the digital payment portal [2] and online banking to e.g. transfer money, book flights, purchase movie tickets and food delivery. In 2011 only about 9.0% of the population had access to the Internet, but the number is proliferating. As of 2019, one in two Nepalese are using the Internet [3], which proves that there is the potential of digitalisation, as one of the main parts of digitalisation is, that people are ready or able to use the technology that business offers.

Many fields have gone digital in Nepal and are doing great. Some of which are among the rental services, such as car rentals, but nothing significant has happened in the renting or subleasing of the apartments. Some businesses with the roots of selling the properties are on the Internet. Subleasing the apartments have also seen the face of the Internet as there are local Facebook pages and groups. Some rental listings are also posted on websites dedicated to second-hand buying and selling. All these factors or potential competitors show that there is a market for an online apartment rental.

The main aim of this thesis was to develop a web-based apartment rental service by creating a business model from the survey, observation and research.

# 2 RESEARCH

Yarshaa DHN is a new concept and potential start-up idea, so, as to get the better perspective on how to get forward, to understanding market necessities and values and to minimize future errors and potential mistakes lots of research-based studies were conducted alongside with a survey in different marketplaces. Existing and potential competitors, legal issues and source of capital was also taken into account.

## 2.1 Housing Nature in Nepal

The ownership of the house is a very general concept in Nepal, about 85.26% of the population resides in their own house.[4] Renting the property (land and apartment) is also considered as one of the major sources of livelihood so, most of the buildings are constructed with renting in mind. The internal migration rate is increasing day by day because of various factors, such as demand for the better job opportunities, job relocation, higher education, quality of life, which also results in a high demand of rental service. In the capital city of Nepal, around 58.65% of people live in a rental apartment[5].

From the tenant point of view, finding the appropriate apartment with required facilities is notably difficult in Kathmandu, one has to settle or compromise quickly because of the risk of losing the property insight and or for some other reasons like shortage of time, resources, personal contacts, security concerns, expenses.

From the landlord point of view, they want to generate income from the vacant apartment as soon as possible. in this context, often they have to compromise on unconventionally lease the apartment, such as splitting the apartment to fit two students even though they wish to have a family who might take the entire apartment.

## 2.2 Digitalisation of Housing

The entire process of renting is done traditionally, which necessitates multiple meetings with multiple potential tenants. This process requires a lot of time and energy. There are few agencies which act as a broker where the landlord has to go and register the house or apartment, which is then followed by phone calls from potential tenants and everything is done manually all over again. Many social media groups are also formed, which facilitates the landlord and tenant to see each-other’s post and listing.

There is a great opportunity in accumulating all the data and listing it online in a web-platform which serves the only purpose of renting apartments. As a tenant, an individual can surf the website visualizing all the available apartment, search the right apartment by sorting the location, the area of the room, number of rooms, available facilities such as parking, playground, balcony, price range. As a landlord, one can post the listing with the wish of a certain amount in rent, a certain type of tenant, such as student, couples, employed, family, whichever suits the apartment type.

As the ability to find the right apartment and right tenant clusters on a single website increases, the traffic eventually grows resulting in a greater service, from which the tenants can spot the right apartment and contact the respective landlord and arrange meetings resulting in an efficient and effective service.

### 2.3 Observation and survey

Data is very important as it can be analysed and several prediction and forecast can be made based upon it. A survey is a reliable method to gather the data. Surveying and analysing the data has always been practised from the dawn of civilization. However, the techniques and methods of surveying have improvised and modernized over the period.

The author exercised an open survey in mass, where a group of people were together. Only a few questions were asked with hypothetical circumstances given to picture. Some survey questions were very simple as, if they would go online and book a room if it ever occurs to them, to some tricky questions, if they would go online just if there is a better offer on more advanced room and if they were willing to change. Another part of the survey was taken with the help of friends where they asked the same questions from their fellow colleagues.

With 76 people taking the survey combined, the majority were tenants (64) and only 12 landlords (figure 1). 48 out of 64 were eager to go online to search the room and 16 were hesitant. Among the 48, 30 of them were willing to go online just to check if there is a better-valued room than the one they are currently residing. The majority of people who were willing to go online appeared young (figure 2).

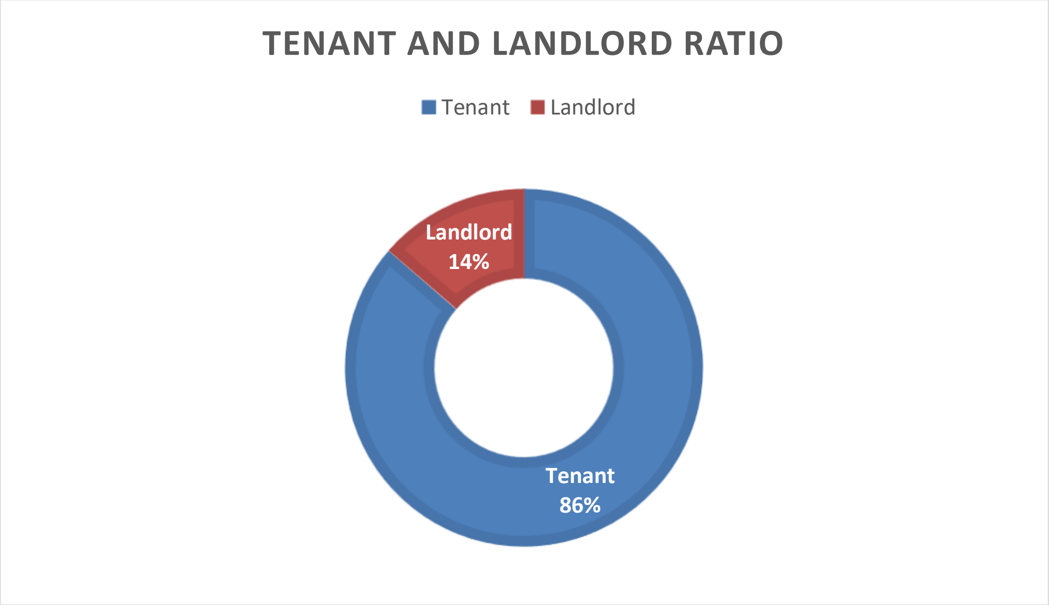


*FIGURE 1. Survey group. Tenant and Landlord ratio.*

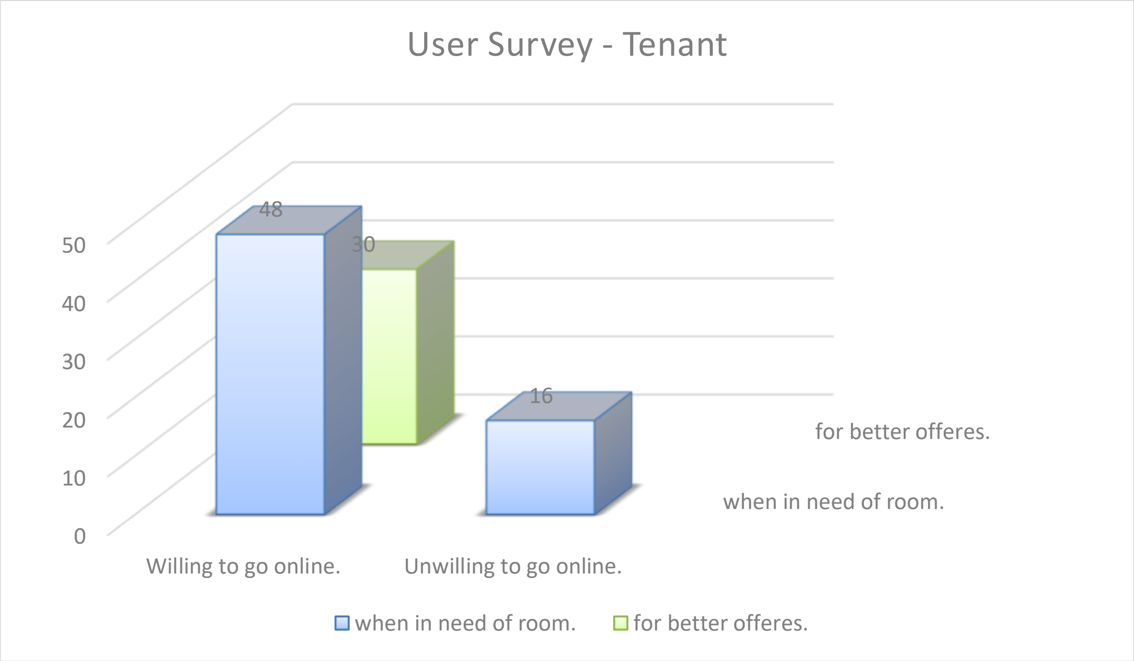


*FIGURE 2. Tenant survey result.*

Among 76 people 12 of them were landlords. About half (7) were ready to go online to post the room. However, when reflected, the right target group of the tenant could be found online and when regarding the possibility of making better offers, 10 of them were willing to go online and post rental.

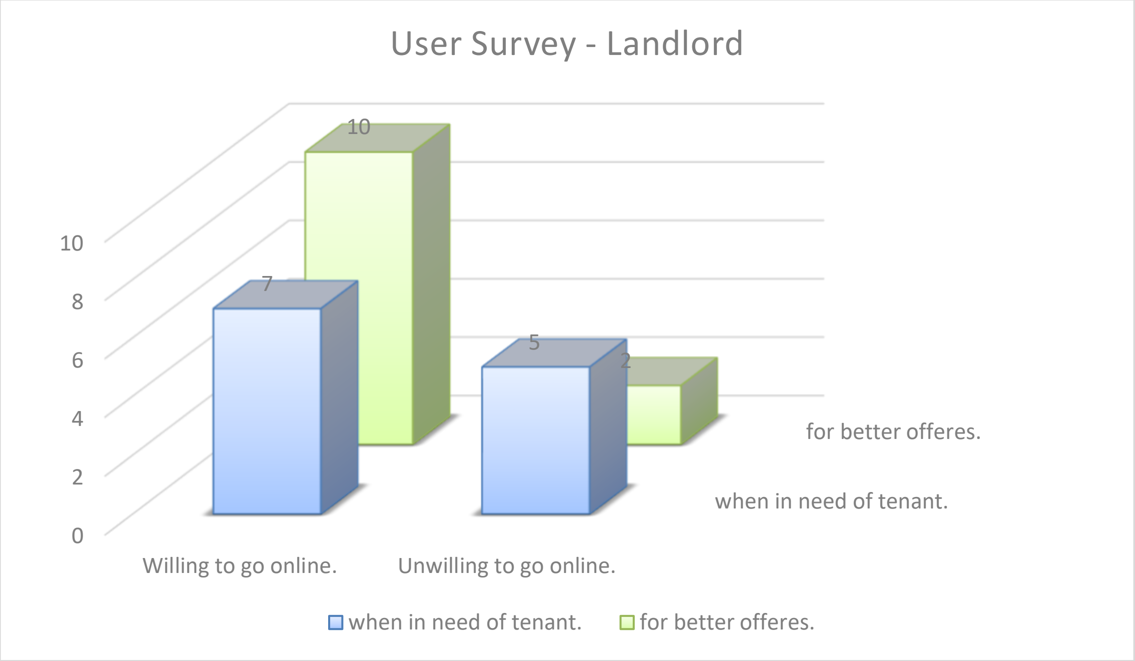


*FIGURE 3. Survey result of landlords.*

### 2.4 Market analysis

From the survey and above figure 1, 2 and 3 it was funnelled that young people who are using the Internet for multiple purposes already, are more willing to adapt the idea. Such group of people and students were taken as the potential target group of the project. Landlords of any age were hesitant of the process of taking a picture, filling the forms and posting it online. However, it was concluded that if a correct marking was done and a hassle-free and straightforward UI and UX are designed, more landlords would be interested in getting in. Furthermore, an assistant can come handy in the case of landlords, where a company staff on request can visit premises and post it for the landlord. This would also be easy to authenticate for the company and administration panel.

# 3 WEB DEVELOPMENT TOOLS

Web development has grown alongside the web. It has also added the scope and responsibility to the developer's shoulder. From the age where developers were expected to perform a specific task and know few technologies in hand to a world where one has to know numerous technologies and perform series of task and of varying scope, the world of programming has changed. This has been only possible because of web development tools becoming advanced. A developer can rely on development tools to debug the problem, test the logic, simulate the interaction and get head-start and leveraging readily available tools and code.

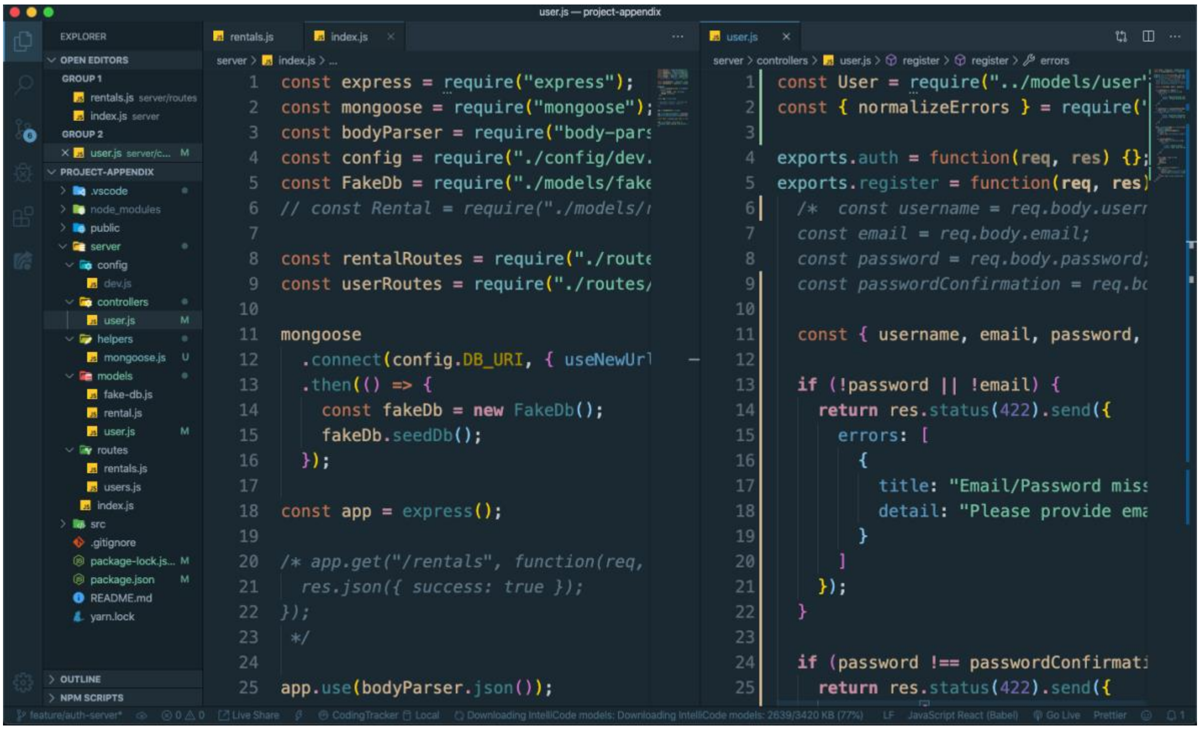
Followings are the web-development tools that were used to make this project possible.

## 3.1 Editor

A code editor is regarded as an essential tool of development as the developer spends most of the time with the code editor during development [6]. So, finding the right code editor is a must.

Visual Studio Code, which is lightweight and fast, was selected as the code editor of choice in this project. It runs in all the major environment, e.g. Windows, macOS and Linux. It readily supports JavaScript [7], which is the programming language used to build the project.

Figure 4 below shows the use of Visual Studio Code as a coding editor in debugging mode.



```
const express = require("express");
const mongoose = require("mongoose");
const bodyParser = require("body-pars
const config = require("./config/dev.
const FakeDb = require("./models/fake
// const Rental = require("./models/r

const rentalRoutes = require("./route
const userRoutes = require("./routes/

mongoose
  .connect(config.DB_URI, { useNewUrl
  .then(() => {
    const fakeDb = new FakeDb();
    fakeDb.seedDb();
  });

const app = express();

/* app.get("/rentals", function(req,
  res.json({ success: true });
});
 */

app.use(bodyParser.json());
```

```
const User = require("../models/user"
const { normalizeErrors } = require('

exports.auth = function(req, res) {};
exports.register = function(req, res)
  /*  const username = req.body.usern
  const email = req.body.email;
  const password = req.body.password;
  const passwordConfirmation = req.bo

  const { username, email, password,

  if (!password || !email) {
    return res.status(422).send({
      errors: [
        {
          title: "Email/Password miss
          detail: "Please provide ema
        }
      ]
    });
  }

  if (password !== passwordConfirmati
    return res.status(422).send({
```

*FIGURE 4. Visual Studio Code in debug mode*

### 3.2 Chrome developer tools

Web development requires continuous tracking of changes, which can be done via a web browser. Most of the modern web-browsers offer developer tools which are specifically designed to help web developers during the development phase. These tools allow a developer e.g. to edit the mark-up and styling of the pages in real-time, to debug the scripting code, and to analyse the performance.

Chrome DevTools is a set of tools that are readily integrated in the web-browser. With it, it is possible to manipulate the DOM, view and change the CSS of the page, the debug script, run the script in the inbuilt console, optimize the website speed and inspect the network activity[8] .Chrome was selected as a browser of choice due to its reliability and excellent developer tools.

Figure 5 below demonstrates the state architecture of the React App under Chrome developer tools.

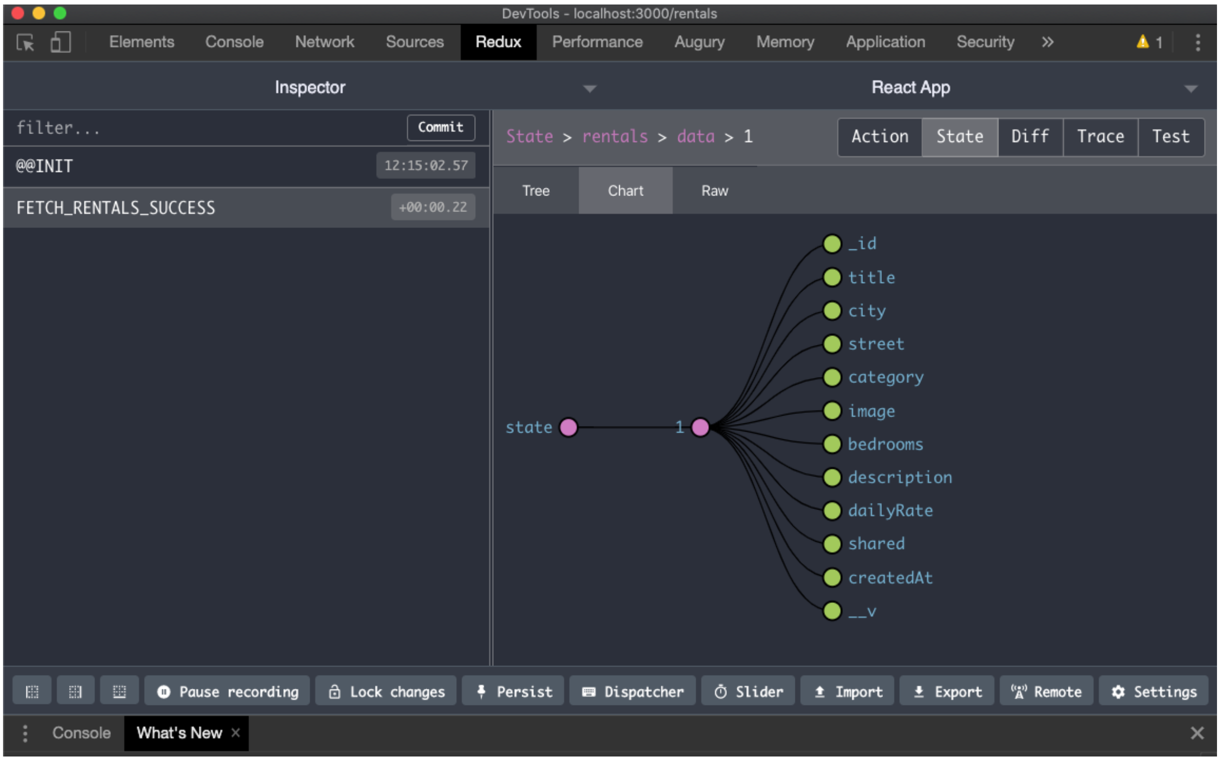


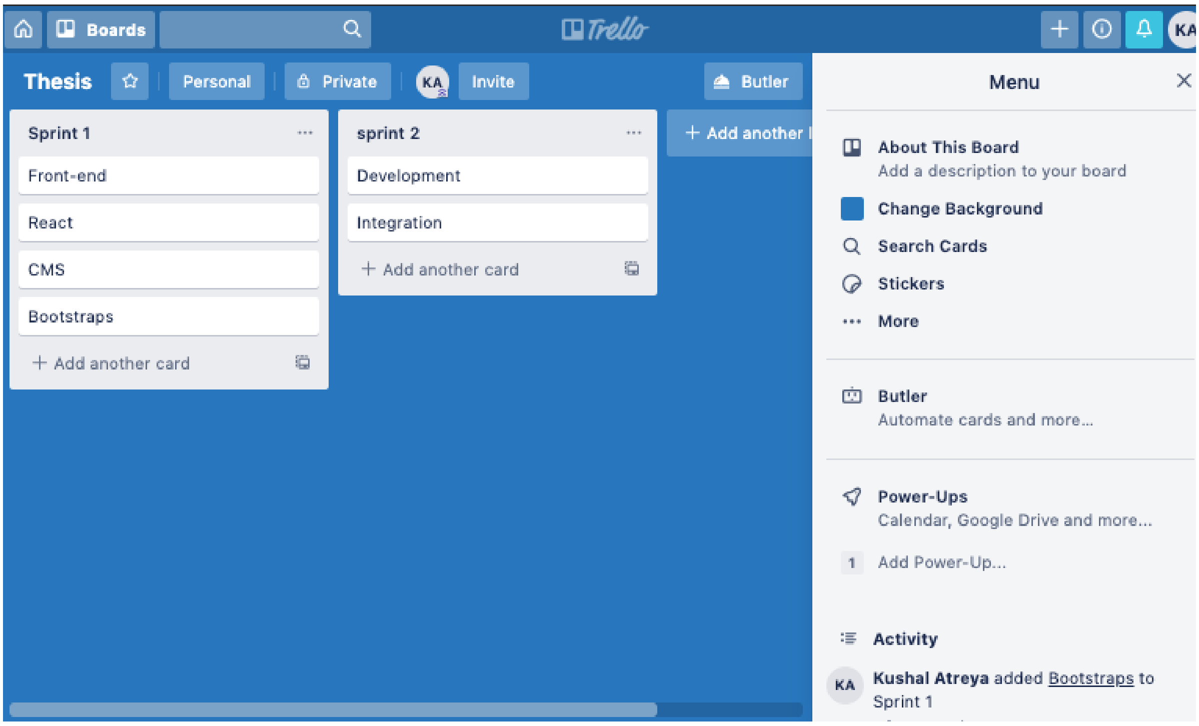
*FIGURE 5. Chrome Developer Tools, showing the chart of state architecture.*

### 3.3 Trello as task manager

Trello is a project management tool, which provides a visual overview of the project. It is excellent in tracking the progress of the project, dividing the project in parts, adding deadlines, upcoming tasks or to having a better picture of the timeline or sprints.

Trello was chosen as the task manager and was used in keeping track of ongoing tasks, upcoming tasks, deadlines, progress reports, project roadmap and task automation.

Figure 6 below shows the use of Trello with a full view of sprint 1 and 2.



Sprint 1

- Front-end
- React
- CMS
- Bootstraps

+ Add another card

sprint 2

- Development
- Integration

+ Add another card

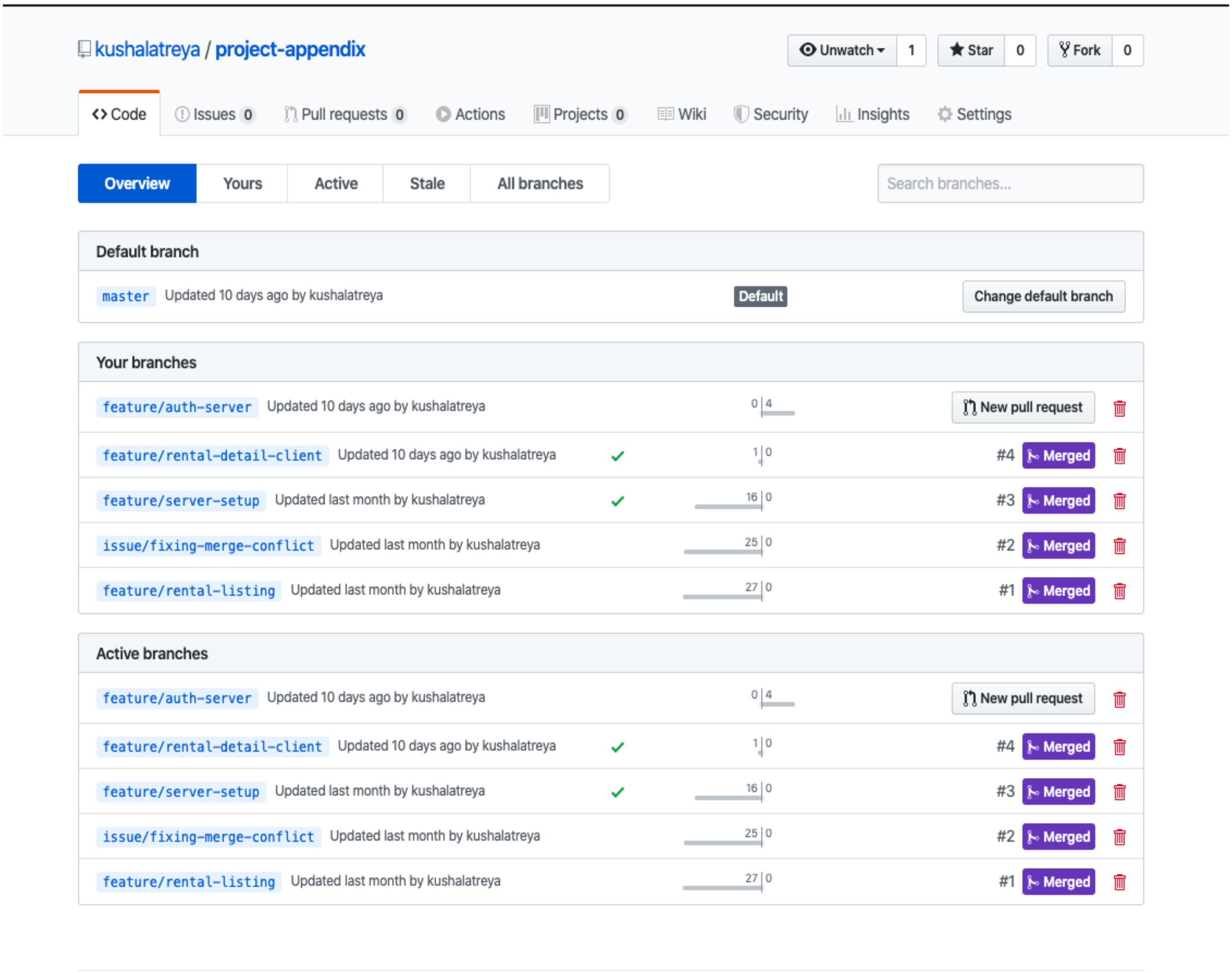
Kushal Atreya added Bootstraps to Sprint 1

*FIGURE 6. Trello showing the sprint.*

### 3.4 Version control system

A version control system is regarded as one of the essential tools in development. It allows a group of developers to work in the same project remotely as every member pushes or uploads the latest code, and everyone can carry it along the way. As every member uploads their code regularly, all the source code is saved and can be accessed from anywhere. This feature is beneficial when the code breaks, crashes the system or creates an unidentified error. The developer can compare the code with the previous version and get a reference or start altogether from the last piece of code that was working.

Git is an open source version control system which features e.g. multiple workflows, branching and staging [9] .GitHub was used as a version control system in this project due to its ease of use, reliability, high storage limit and free private repositories.



Default branch

| | | | |
|---|---|---|---|
| master | Updated 10 days ago by kushalatreya | Default | Change default branch |

Your branches

| | | | | | |
|---|---|---|---|---|---|
| feature/auth-server | Updated 10 days ago by kushalatreya | | 0 \| 4 | New pull request | |
| feature/rental-detail-client | Updated 10 days ago by kushalatreya | ✓ | 1 \| 0 | #4 | Merged |
| feature/server-setup | Updated last month by kushalatreya | ✓ | 16 \| 0 | #3 | Merged |
| issue/fixing-merge-conflict | Updated last month by kushalatreya | | 25 \| 0 | #2 | Merged |
| feature/rental-listing | Updated last month by kushalatreya | | 27 \| 0 | #1 | Merged |

Active branches

| | | | | | |
|---|---|---|---|---|---|
| feature/auth-server | Updated 10 days ago by kushalatreya | | 0 \| 4 | New pull request | |
| feature/rental-detail-client | Updated 10 days ago by kushalatreya | ✓ | 1 \| 0 | #4 | Merged |
| feature/server-setup | Updated last month by kushalatreya | ✓ | 16 \| 0 | #3 | Merged |
| issue/fixing-merge-conflict | Updated last month by kushalatreya | | 25 \| 0 | #2 | Merged |
| feature/rental-listing | Updated last month by kushalatreya | | 27 \| 0 | #1 | Merged |

*FIGURE 7. GitHub with branches.*

Figure 7 above shows the use of GitHub as the version control system with the branches. The branches represent the smaller functionalities of the code they represent. Active branches show the branch where the work is still in progress or not ready to be merged.

# 4 MODERN WEB APPLICATION

In correspondence to the Internet, a web application has seen a revolutionary growth. From a static informative website to a website filled with multimedia and animation to a now mobile-friendly, responsive, scalable and single-page concept of today. The era and notion of the modern web application is shifting very rapidly.

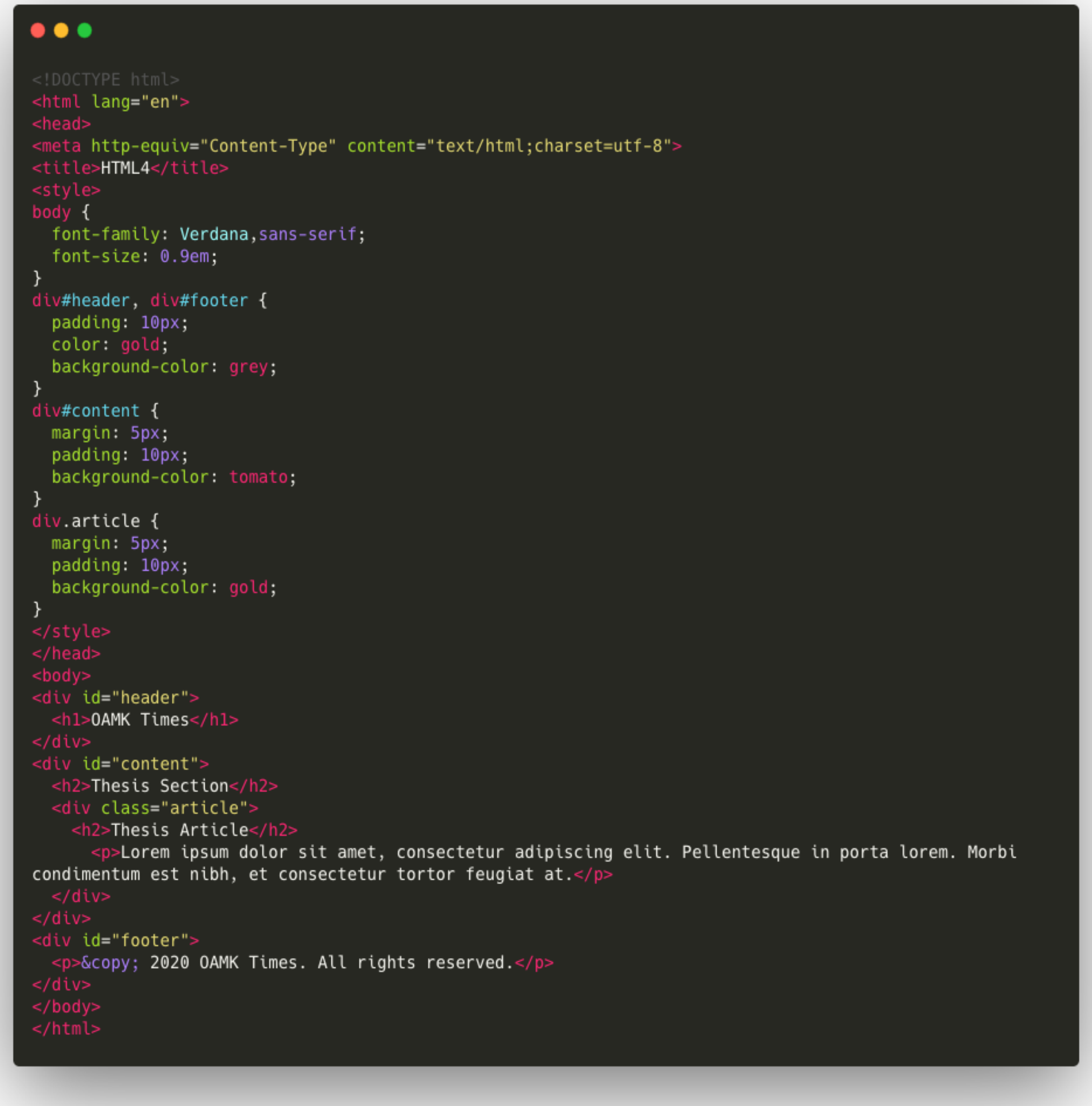
The web-architecture, which was deemed as modern and futuristic a few years ago, is now far behind the par. If a web-application is not updated and upgraded continuously, it might become obsolete in a quick span of time. The web-technologies, which were used in this application, are described concisely below.

## 4.1 Client-side web technologies

The following client-side web-technology was used for developing the application prototype.

### 4.1.1 HTML

HTML stands for the Hypertext Markup Language. It  gives a structure to the web page. HTML elements are described by tags. Each tag is understood by the browsers and rendered collectively by giving a complete structure to the web-page. The browser does not display HTML tags, but it uses them to determine the layout of the document[10]. HTML can embed CSS for the styling and JavaScript for the added functionality.

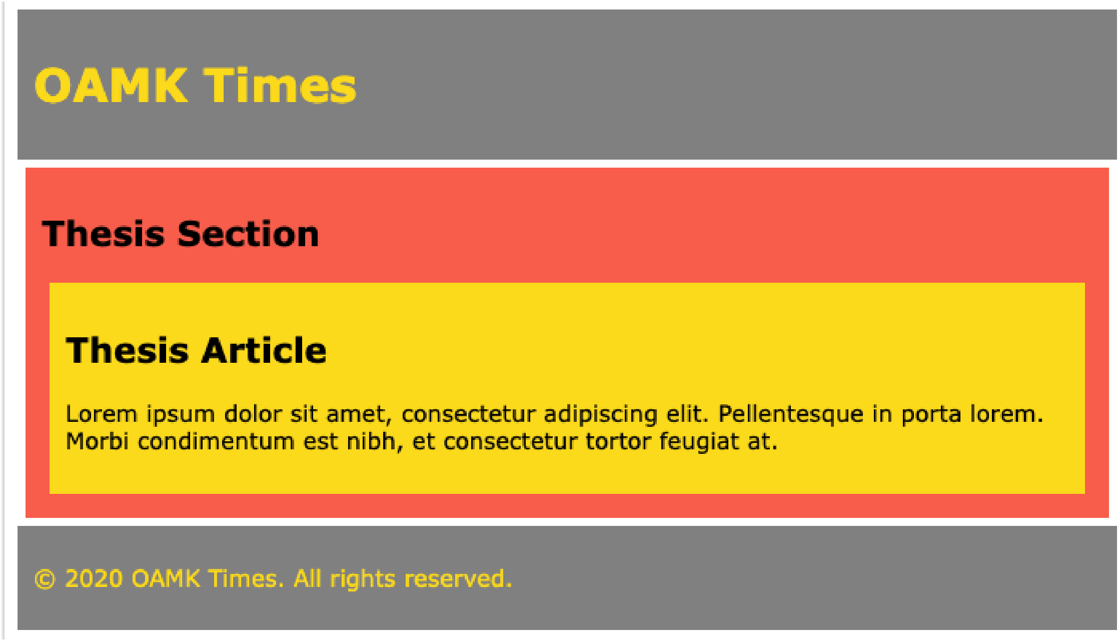


```
<!DOCTYPE html>
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html;charset=utf-8">
<title>HTML4</title>
<style>
body {
  font-family: Verdana,sans-serif;
  font-size: 0.9em;
}
div#header, div#footer {
  padding: 10px;
  color: gold;
  background-color: grey;
}
div#content {
  margin: 5px;
  padding: 10px;
  background-color: tomato;
}
div.article {
  margin: 5px;
  padding: 10px;
  background-color: gold;
}
</style>
</head>
<body>
<div id="header">
  <h1>OAMK Times</h1>
</div>
<div id="content">
  <h2>Thesis Section</h2>
  <div class="article">
    <h2>Thesis Article</h2>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem. Morbi
condimentum est nibh, et consectetur tortor feugiat at.</p>
  </div>
</div>
<div id="footer">
  <p>&copy; 2020 OAMK Times. All rights reserved.</p>
</div>
</body>
</html>
```

*FIGURE 8. HTML Document with CSS for styling.*

As shown in figure 8,  when the browser receives the document, it recognises the file as HTML or more specifically HTML 5 after reading the tag <!DOCTYPE html>. <html> is the root element tags which holds all other tags. <head> consists of all the metadata like title, CSS and import links.<body> consists of all the UI part of the page.

Figure 9 below shows the HTML document that is rendered in the browser. It is the output of the code which is shown in figure 8 above.



OAMK Times

Thesis Section

Thesis Article

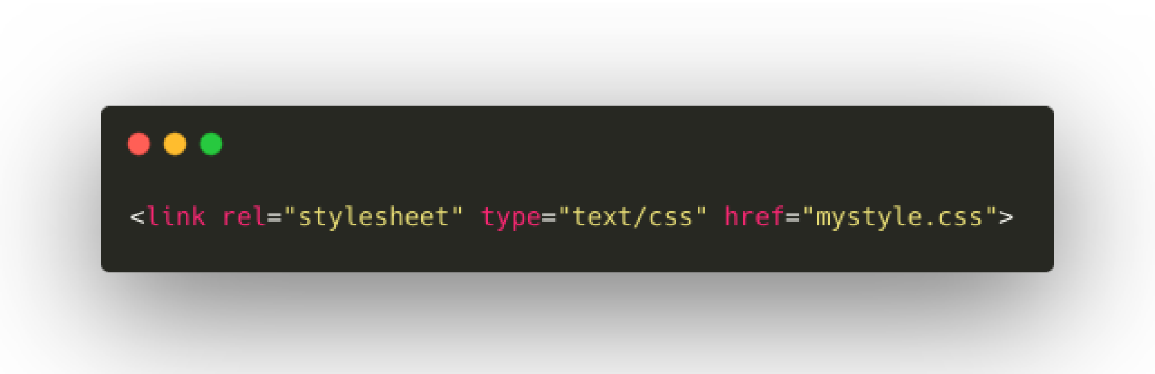
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem. Morbi condimentum est nibh, et consectetur tortor feugiat at.

© 2020 OAMK Times. All rights reserved.

*FIGURE 9. HTML Document rendered in the browser.*

### 4.1.2 CSS

CSS stands for the Cascading Styling Sheet. The primary responsibility of CSS is to provide styling to the HTML element. CSS can be written in a different file and imported in the HTML document as shown in figure 10.

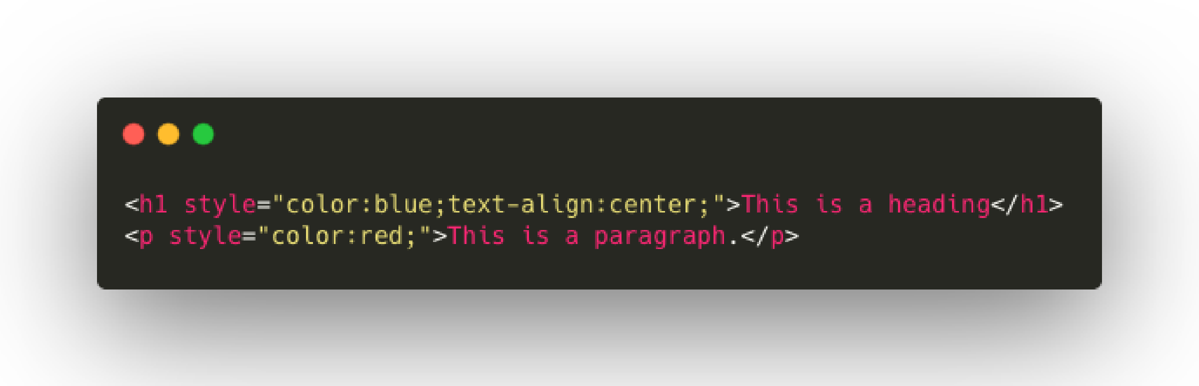


```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

*FIGURE 10. CSS import from an external link.*

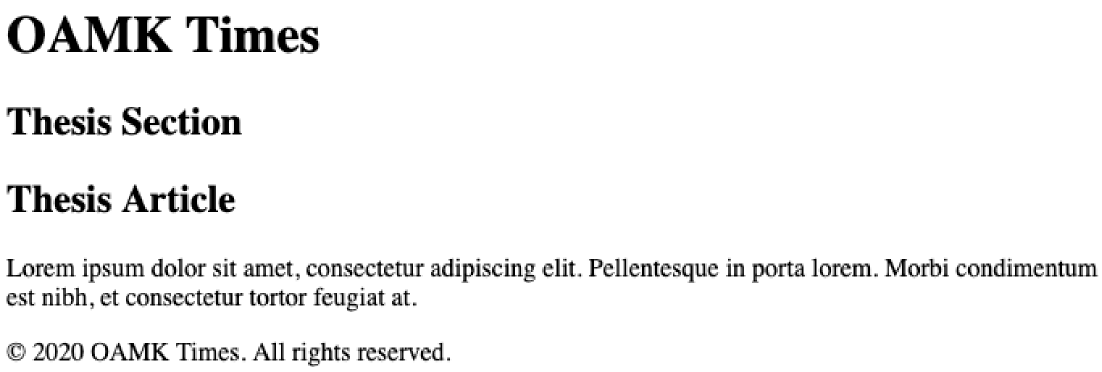
CSS can be written in the same file as in HTML, as shown in Figure 8,  and in the same line of the HTML tags, which is referred to as an inline CSS. The use of Inline CSS is demonstrated in the figure 11.

CSS gives the visual feel of the website and power to showcase each element in an intended manner. It has arrays of in-built styling syntax for setting the colour, margin, border, position, width, height, which stretches to navigation, tables, form, list to more advanced 2D, 3D transformation, gradients, media-queries, viewport to now in a responsive or mobile-friendly world with CSS-Grid and Flex-Box.



```
<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```

*FIGURE 11. Inline CSS.*



# OAMK Times

## Thesis Section

## Thesis Article

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem. Morbi condimentum est nibh, et consectetur tortor feugiat at.

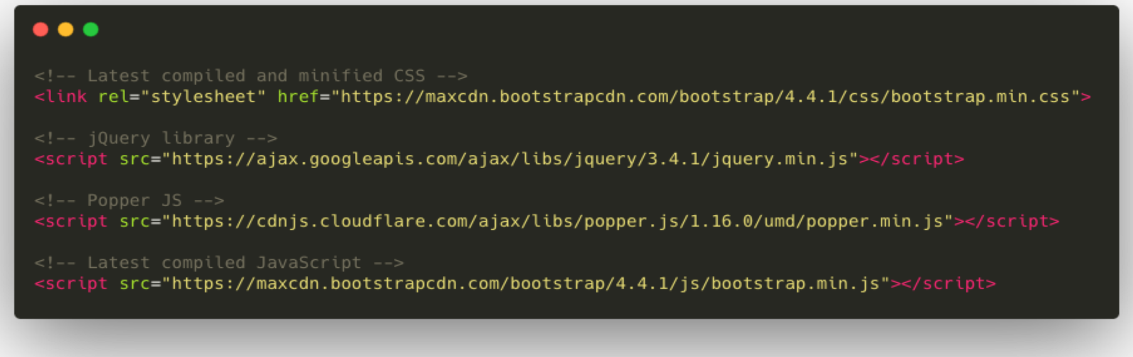
© 2020 OAMK Times. All rights reserved.

*FIGURE 12. HTML Document with no CSS.*

Figure 12 above is the outcome of the same HTML document as in Figure 8, but without CSS, which explicates how significant CSS is.

### 4.1.3 Bootstraps

Bootstraps is the most popular front-end framework to design responsive web-pages[11]. Because of ease of use, responsiveness and browser compatibility, Bootstraps is very popular. Bootstraps can be used just by providing CDN as in the figure 13, after which Bootstraps' readily available syntax can be implemented. Bootstraps has a readily available syntax for most-use cases, such as tables, buttons, progress-bar, buttons, cards, dropdowns, forms, navbar, header, footer and carousel, which allows users to head-start without concerning about responsiveness.



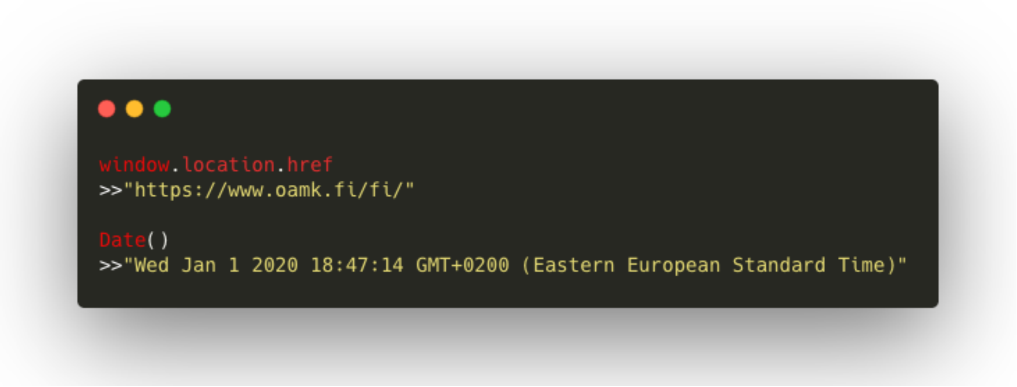
```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<!-- Popper JS -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
```

*FIGURE 13. Bootstrap CDN.*

### 4.1.4 JavaScript

JavaScript is one of the most widely used programming languages[12]. It is a scripting language where a code snippet can be executed without the whole file being compiled. JavaScript was introduced as a front-end or client-side language, but with time, JavaScript has grown to be a full-fledged programming language with the addition of node as a server-side language. Furthermore, JavaScript is now also used in Machine Learning, Industrial IoT, AI and Robotics.

Due to the flexibility of use in both front-end and back-end, the author has chosen JavaScript as the choice of programming language.



```
window.location.href
>>"https://www.oamk.fi/fi/"

Date()
>>"Wed Jan 1 2020 18:47:14 GMT+0200 (Eastern European Standard Time)"
```

*FIGURE 14. JavaScript Method to get URL of the website and current date.*

figure 14 above shows the power and ease of JavaScript. One line method can display the URL of the website and current date.

## 4.2 Single Page Application

Single Page Application, also referred to as SPA is the modern form of web-application. When the page loads, only the required content is downloaded from the server, which makes the web-experience light and fast. When a user interacts with the web-page, It uses Ajax and DOM updates to dynamically update in a responsive way without the wait of re-loading. This approach takes most of the work under the hood of client-side.

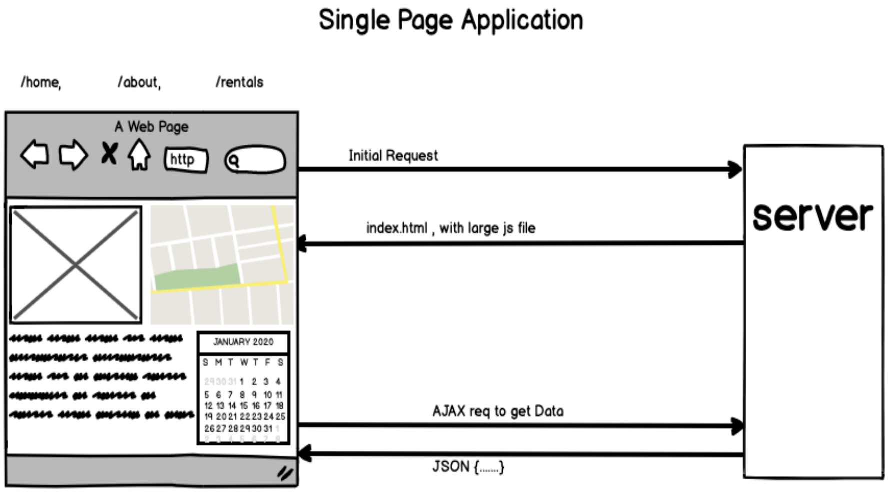


*FIGURE 15. Work Flow of Single Page Application.*

Figure 15 above demonstrates the workflow of the Single Page Application. With the use of React JavaScript's Virtual DOM and React-Router, once the page loads, it can handle the changes and interaction much quicker.

### 4.3 Multi Page Application

Multi-page is an old approach, as shown in the below figure 16, where all the user interaction leads to the server request, and a new page is rendered. This type of application has a longer loading time and it consumes more data as every sub-page needs a refresh and a server request. It has layers of complexities and it is challenging to develop on top of it. It is slow and maintaining and updating the application is time-consuming.

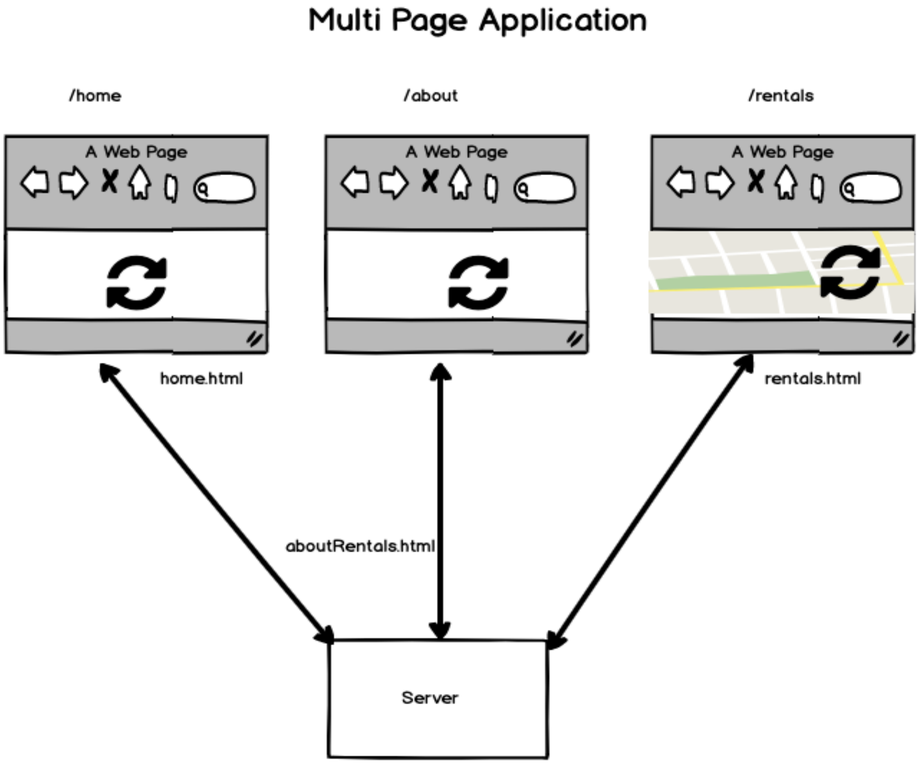


*FIGURE 16. Work Flow of multi-page application.*

# 5 DESIGN AND DEVELOPMENT

Yarshaa DHN is a full-stack single page web-application leveraging modern libraries and programming languages. It uses JavaScript extensively with React in front-end and Express in the backend. The responsiveness and mobile-friendly design concept is kept coherent throughout the development process.

## 5.1 Requirements

A working and functioning prototype was aimed for this thesis project which could be carried on and continued. Upon investment and hands from people of different fields, it could grow as a start-up. For this prototype, user stories were created, keeping in mind the purpose of the application and target group views.

User stories are listed below:

- As a user, I would like to view the rentals without logging in.
- As a user, I would like to have the secure user-login feature.
- As a user, I would like to see the location of the rentals.
- As a user, I would like to see the available facilities and a description of the rentals.
- As a user, I would like to have search functionality.
- As a user, I would like to have the ability to book the rentals.
- As a landlord, I would like to have my user account.
- As a landlord, I would like to upload my rentals.
- As a landlord, I would like to have the ability to accept or reject offers.

Taking user-stories into account, a road map was created to capture the relation of how components and web-pages would flow, which can be seen in below figure 17 below.

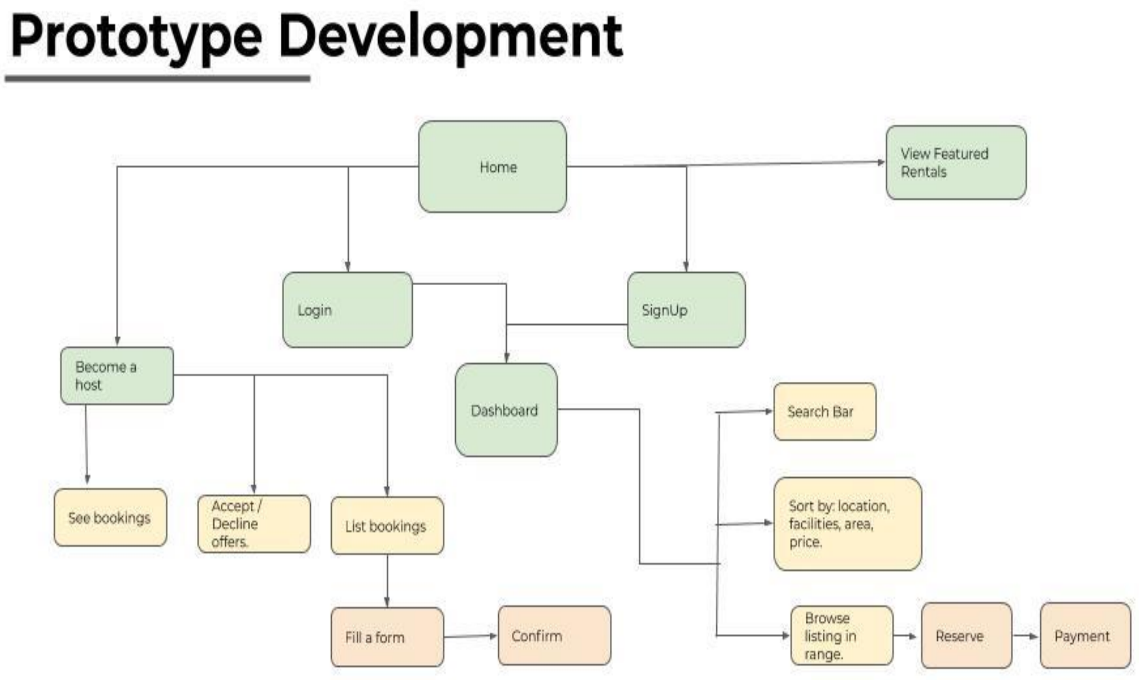


*FIGURE 17. Prototype Development.*

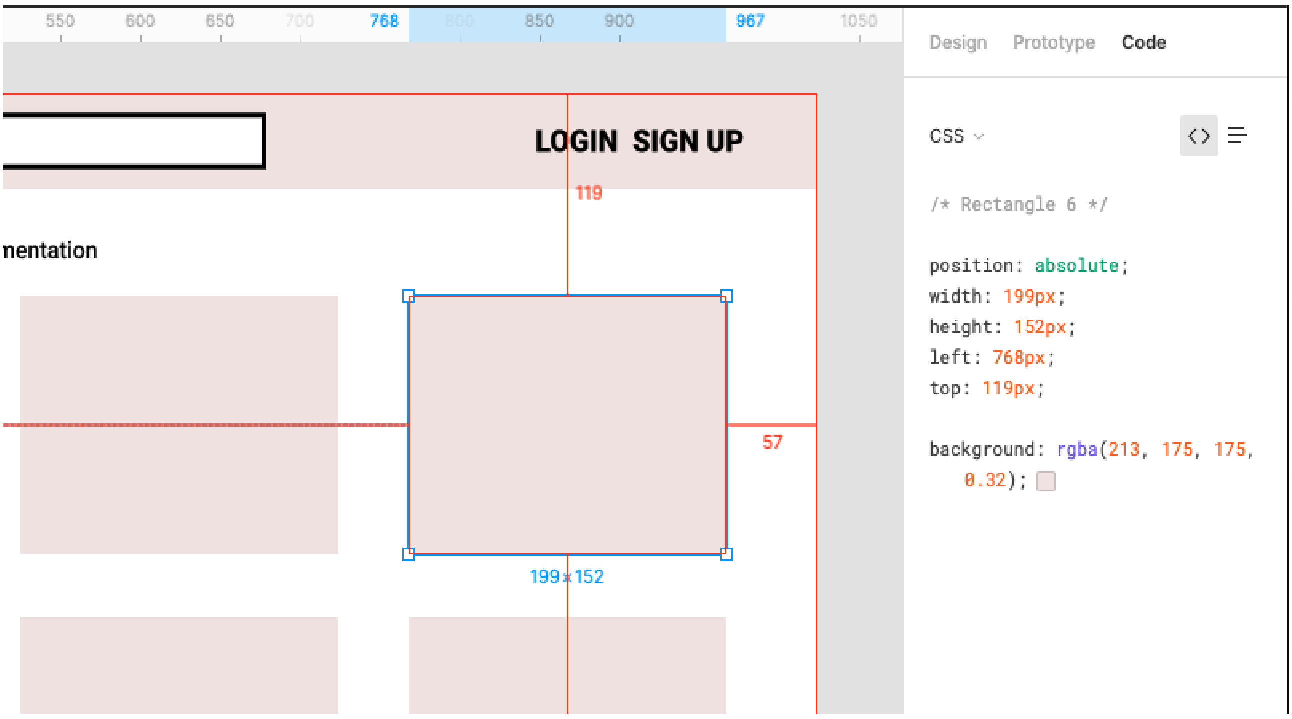
### 5.2 Frontend

Frontend is the interactive layer which sits above the back-end. It is responsible for the overall feel and experience of the user. Everything the user visualises or can visualise and interact is under front-end.

Below are the technologies that were used in the front-end section of the application.

#### 5.2.1 Figma

Figma is a free cloud-based design tool which is used to create mock-up design and wireframe of the website. It is generally used by the designer and passed to the front-end developer. The Front-end developer develops the web-page taking the design of the wireframe into account. Figma is component-based, the re-useable part is created as a component and used multiple times. Figma allows copying the CSS code snippet, which gives the head-start to the developer, which is shown in the figure 18.



```
/* Rectangle 6 */

position: absolute;
width: 199px;
height: 152px;
left: 768px;
top: 119px;

background: rgba(213, 175, 175,
    0.32);
```

*FIGURE 18. Figma as design tool showcasing the CSS of selected rectangle-component.*

The wireframe of the application was designed using the Figma, some of which are listed below in the figures 19 and 20.

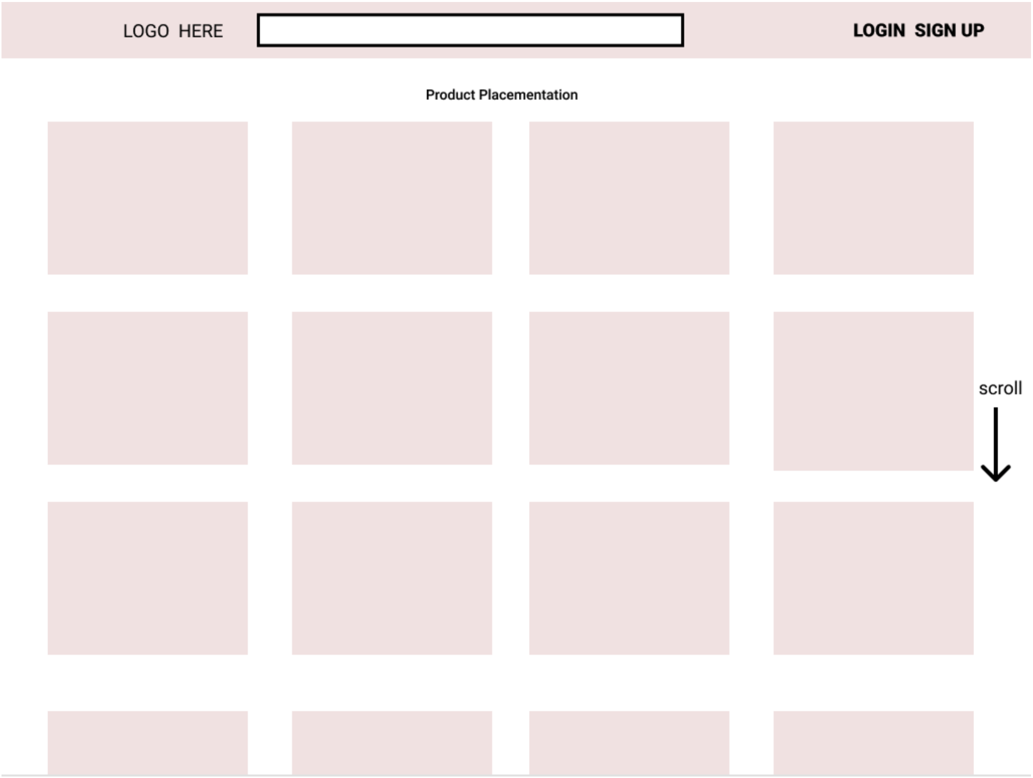


*FIGURE 19. Wireframe design of the dashboard.*

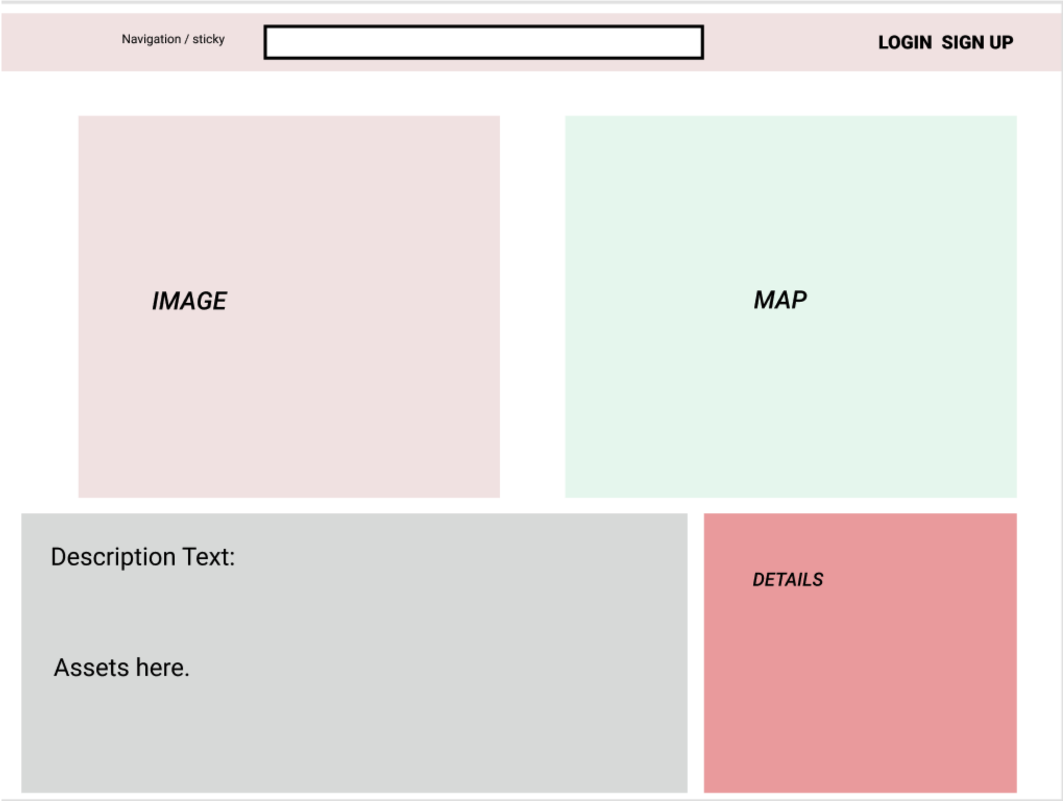
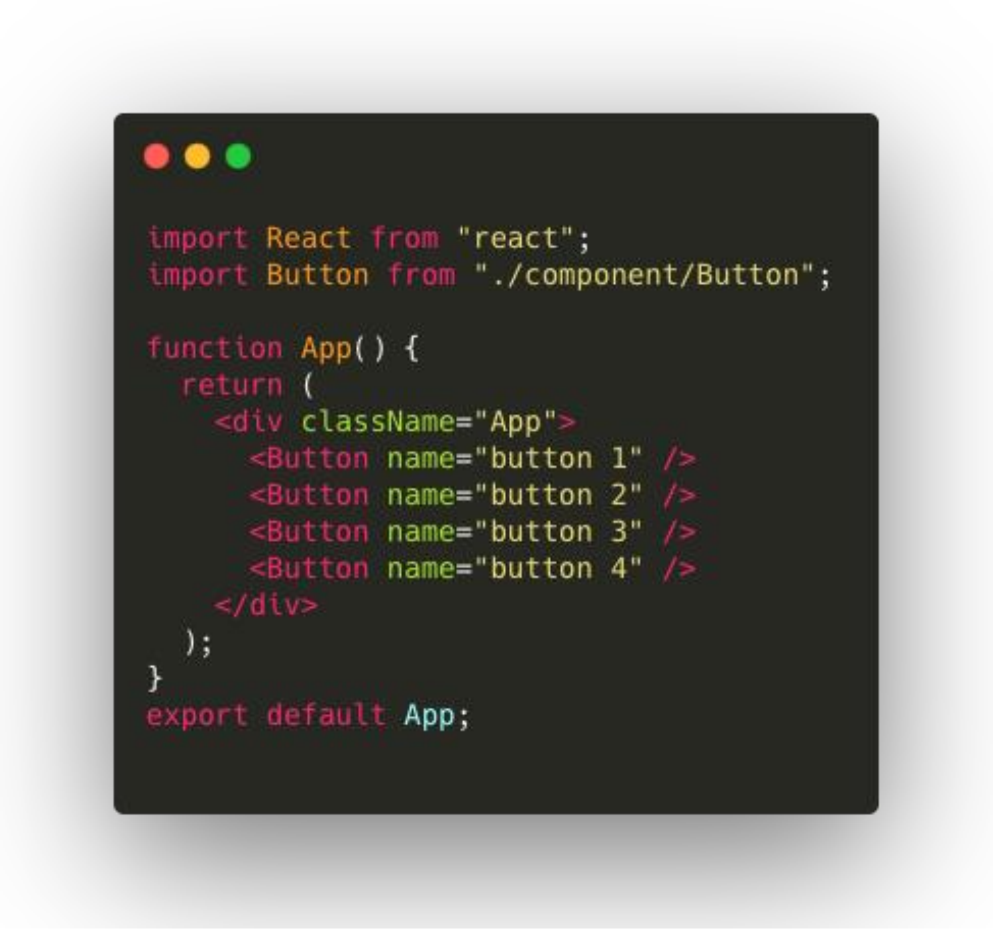


*FIGURE 20. Wireframe of the detail component page.*

### 5.2.2 ReactJS

React is a front-end open-source JavaScript library which can create interactive and modular UI reusable-components[13]. React elegantly updates the JavaScript DOM and renders the correct component with user interaction. It uses Virtual-DOM which captures the current state of the page and only re-renders the parts of the component that has changed. This approach delivers a smooth user-experience.

React is component-based, components are independent and reusable. In the application, a button can be a component, and the same component can be used or called whenever a button is required. This practice reduces the lines of code, embraces clean code, and massively reduces the effort. The components are building blocks of the application, which is arranged to form the layout of the application.



```
import React from "react";
import Button from "./component/Button";

function App() {
  return (
    <div className="App">
      <Button name="button 1" />
      <Button name="button 2" />
      <Button name="button 3" />
      <Button name="button 4" />
    </div>
  );
}
export default App;
```

*FIGURE 21. Using same Button component multiple times.*

Figure 21 above shows the working model of the react. It imports the essential component, which is built for the application, and renders it as many times it is to be used. The component (button) takes a name as a props which can be set dynamically, allowing to use the same component multiple times with the dynamic name.

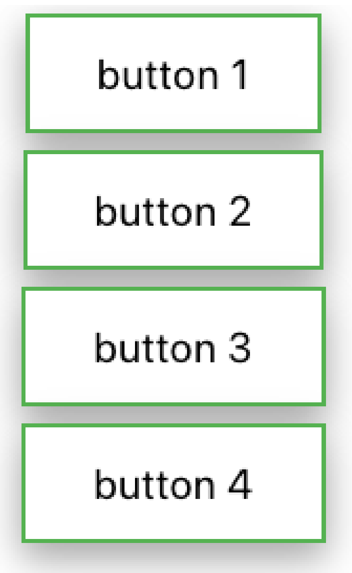


*FIGURE 22. Rendered component with dynamically set names.*

Figure 22 shows how the rendered component with the dynamically set names is shown in the output.

The application is built using React, and the practice of creating the component is extensively leveraged, which is also demonstrated in the below figure 23.

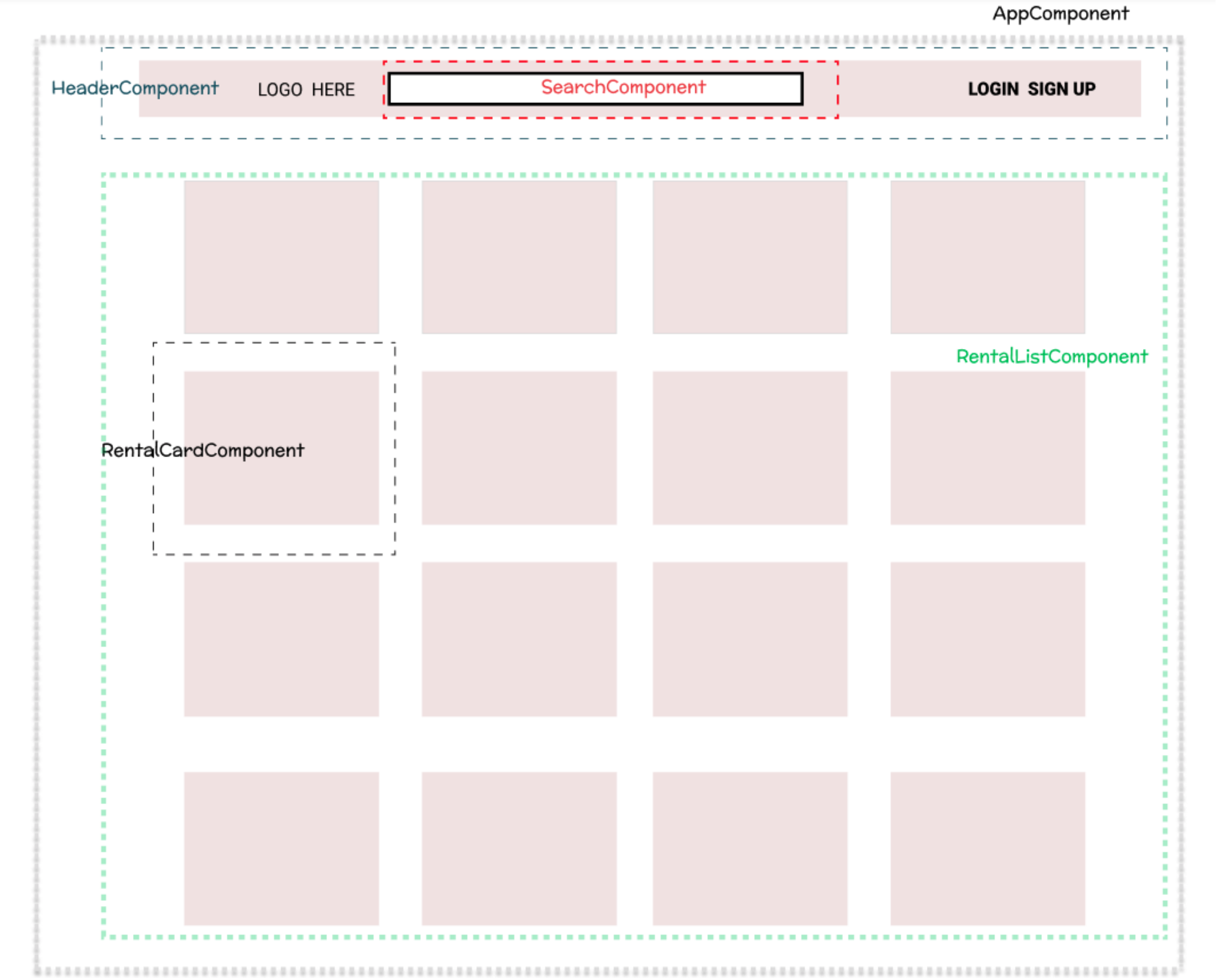


*FIGURE 23. Component Design of the Application*

### 5.2.3 Redux

Redux helps in writing applications that behave consistently, run in different environments (client, server, and native) and are easy to test[14]. It is an open-source and very small JavaScript library used to manage the state. Redux is mostly associated with React, but it can be used with other frameworks, such as Angular and Vue JS. The Redux state is read-only. It stays in the root of the application and maintains the state as a single immutable state, it is read-only, and the action should be dispatched if the state needs modification. An action is a plain JavaScript object, which is created by action creators. It acts as the events as it sends data from the user interaction to the store.

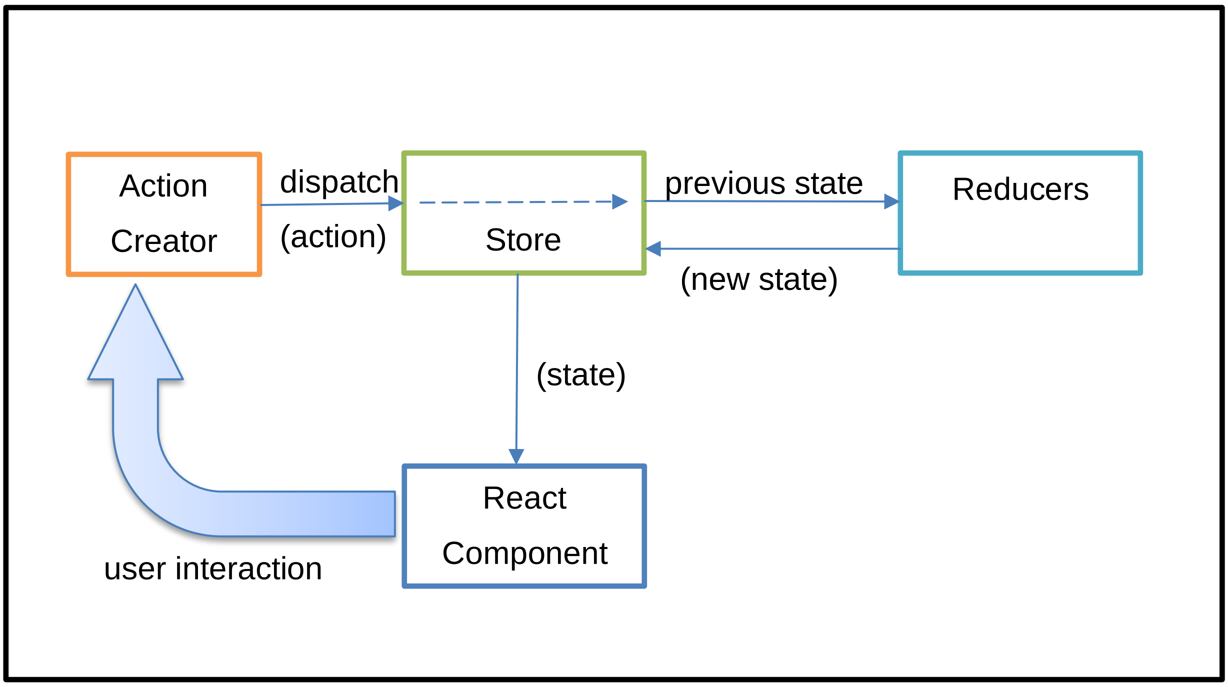


*FIGURE 24. Redux architecture*

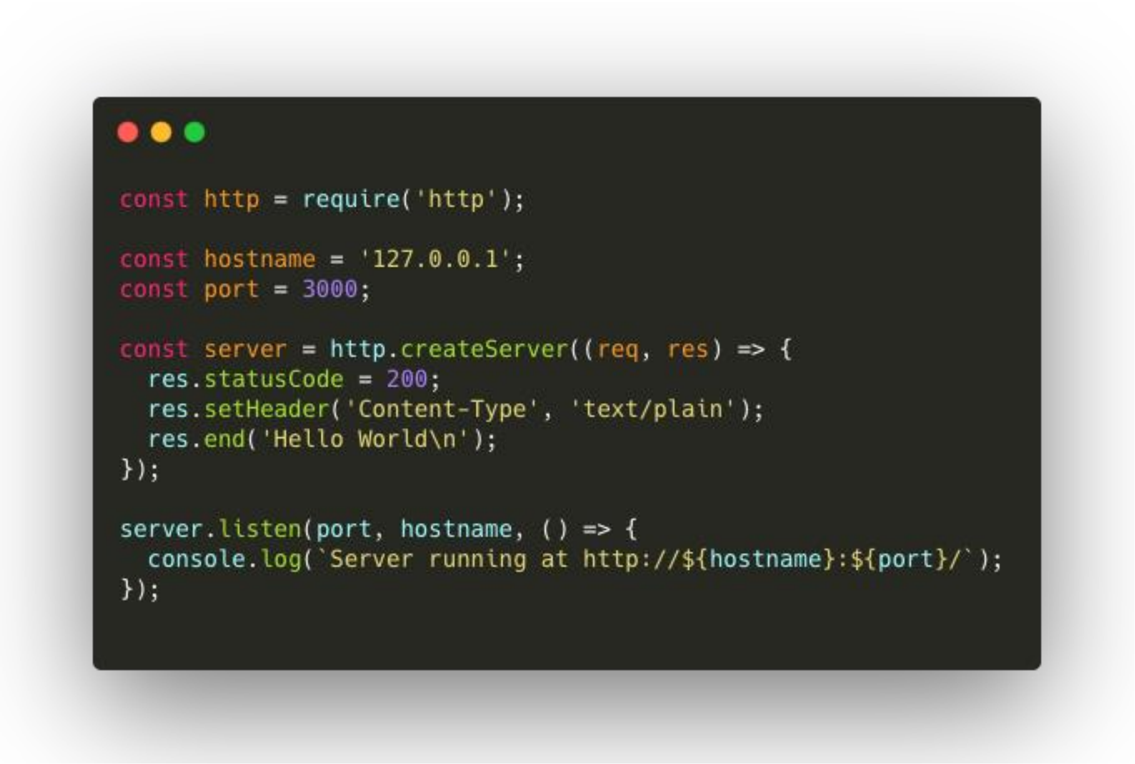
Redux is very predictable as it makes the use of the state very straight forward and having a predictable result makes maintenance practical. Figure 24 above demonstrates the architecture of Redux.

### 5.3 Back-End Technology

The back-end is part of application-script which runs in the background without user knowledge. It is responsible for receiving the data from the users, handling it and storing it in the server. The Back-end part embeds the UI of the websites, which enables the users to request and receive the information from the server and in general interact with the website.

### 5.3.1 Node.js

Node.js is an open-source server-side JavaScript runtime environment. It is cross-platform as it runs in various platforms such as Windows, Linux, Mac OS. It is a single-threaded event loop and very memory efficient, which eliminates the waiting time by using non-blocking I/O calls. It is used to make utilities and or a web server. It is best suited for I/O bound applications, data streaming applications, data intensive and real-time applications, JSON APIs based applications and single page applications[15]. Figure 25 shows the HTTP server with node.js. Many connections can be handled and upon each request a callback is triggered. If there is no request, Node.js will sleep.

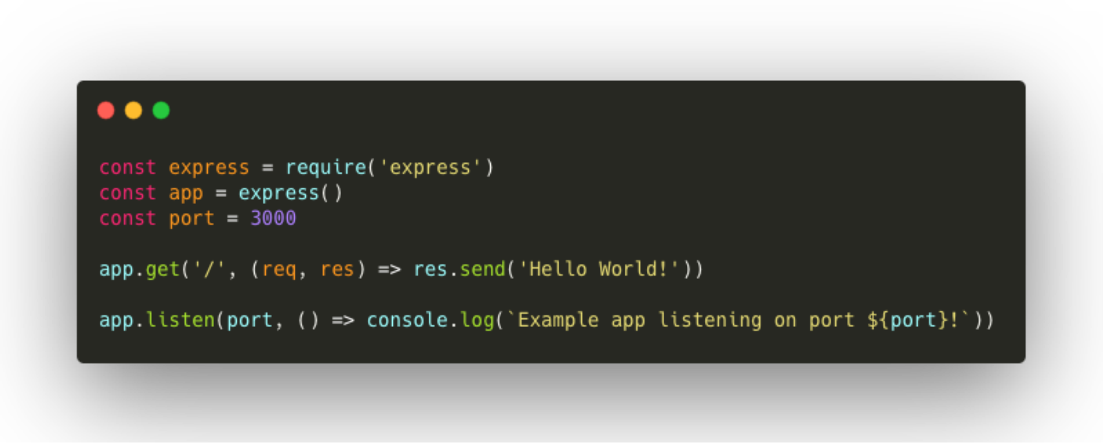


```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

*FIGURE 25. HTTP server with node.js*

### 5.3.2 Express

Express is the most popular Node web framework. It has a great developer community. There are compatible middleware packages for almost any web development problem. There are libraries to work, e.g. with the cookies, URLs, session requests, login, authentication [16]. It is a modular framework built on top of Node.js. It presents more functionality and provides the developer with a head-start by providing the equivalent outcome while writing fewer lines of code. It is a lightweight framework to create the application in the MVC architecture on the server-side.



```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => res.send('Hello World!'))

app.listen(port, () => console.log(`Example app listening on port ${port}!`))
```

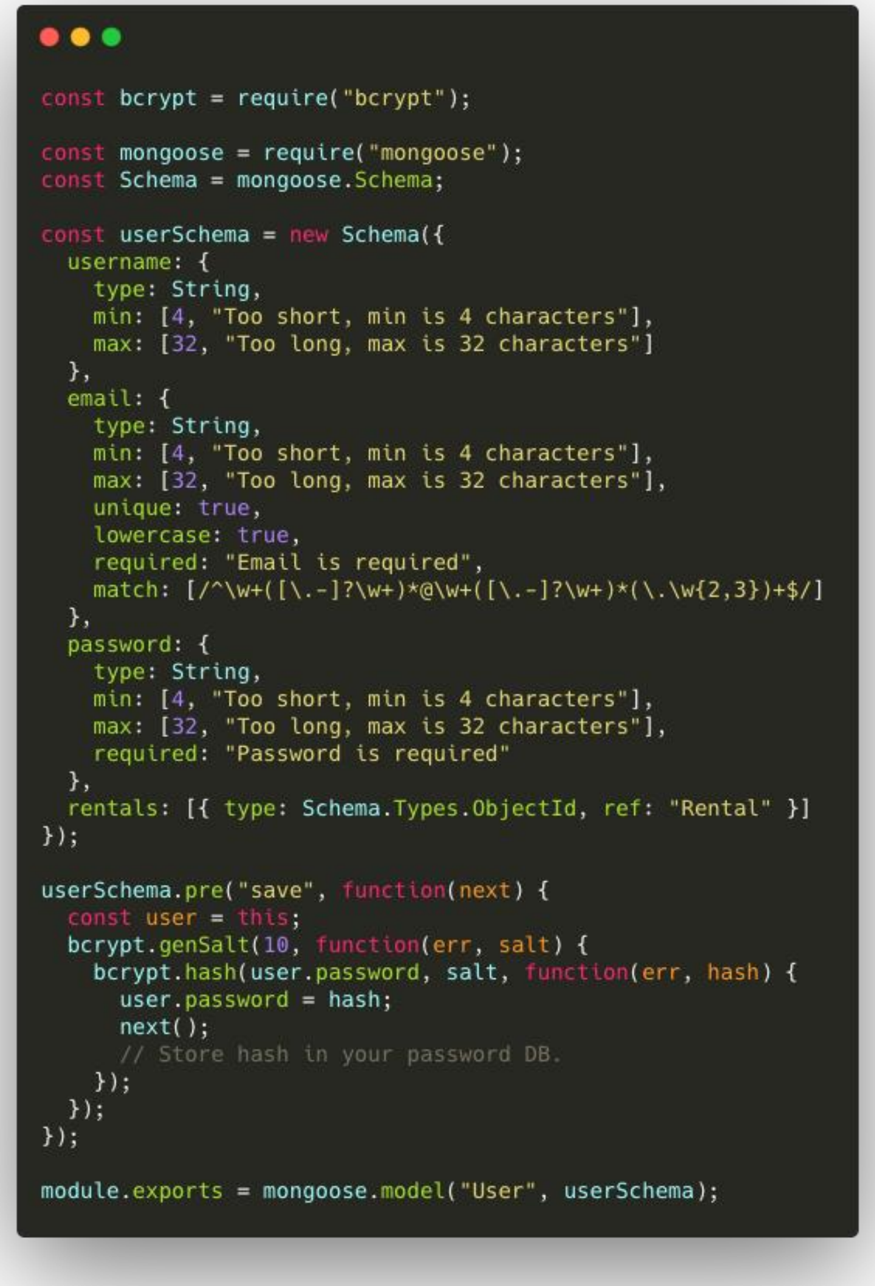
*FIGURE 26. Express app.*

Figure 26 shows the express app which responds with “Hello World!” when the request is made to the root URL.

### 5.3.3 MongoDB

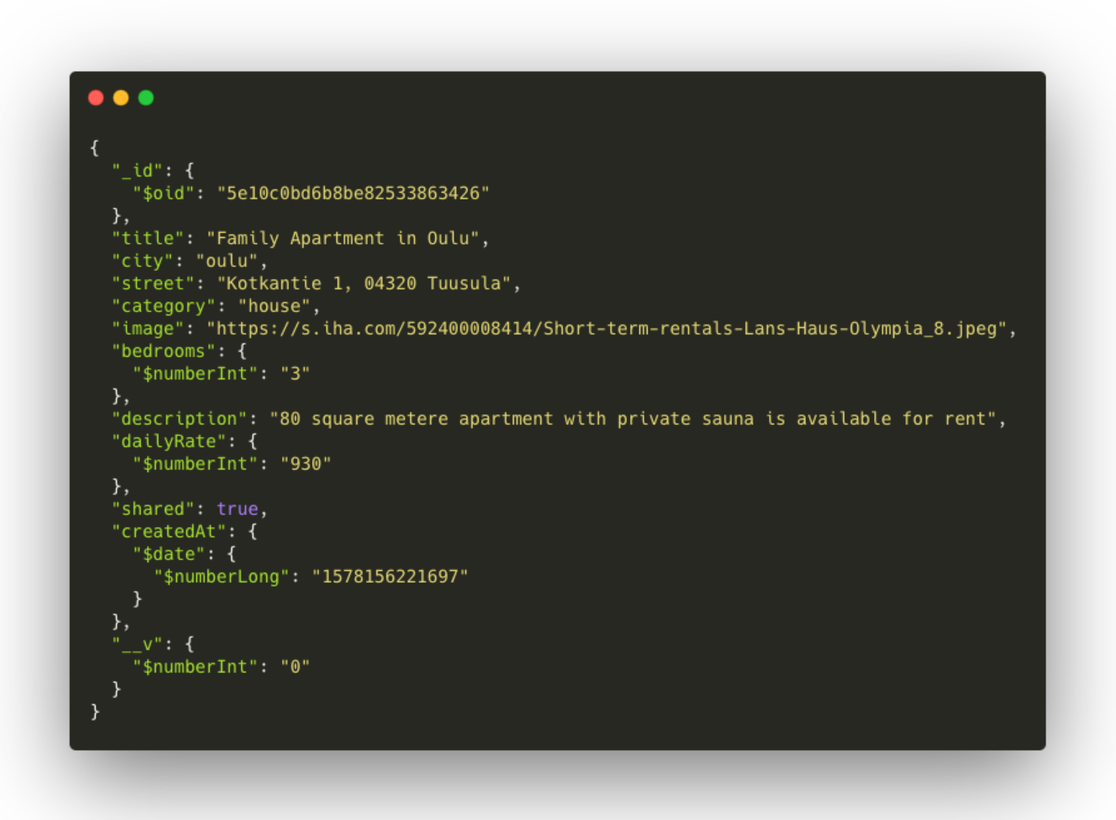
The database is used to store the data online. Data is saved in a specific format or model or schema as shown in the figure 27. When in need, data can be searched, sorted and presented to the client-side.

MongoDB was used in the application because of its ease of use, high performance, scalability and geospatial support. It is a NoSQL database which uses the JSON format to store the document instead of the table format as shown in below figure 28. It provides a high performance and scalability along with data modelling and management of large sets of data with an auto-scaling feature[17].



```
const bcrypt = require("bcrypt");

const mongoose = require("mongoose");
const Schema = mongoose.Schema;

const userSchema = new Schema({
  username: {
    type: String,
    min: [4, "Too short, min is 4 characters"],
    max: [32, "Too long, max is 32 characters"]
  },
  email: {
    type: String,
    min: [4, "Too short, min is 4 characters"],
    max: [32, "Too long, max is 32 characters"],
    unique: true,
    lowercase: true,
    required: "Email is required",
    match: [/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/]
  },
  password: {
    type: String,
    min: [4, "Too short, min is 4 characters"],
    max: [32, "Too long, max is 32 characters"],
    required: "Password is required"
  },
  rentals: [{ type: Schema.Types.ObjectId, ref: "Rental" }]
});

userSchema.pre("save", function(next) {
  const user = this;
  bcrypt.genSalt(10, function(err, salt) {
    bcrypt.hash(user.password, salt, function(err, hash) {
      user.password = hash;
      next();
      // Store hash in your password DB.
    });
  });
});

module.exports = mongoose.model("User", userSchema);
```

*FIGURE 27. Schema for saving a user group by salting and encrypting the password in MongoDB.*



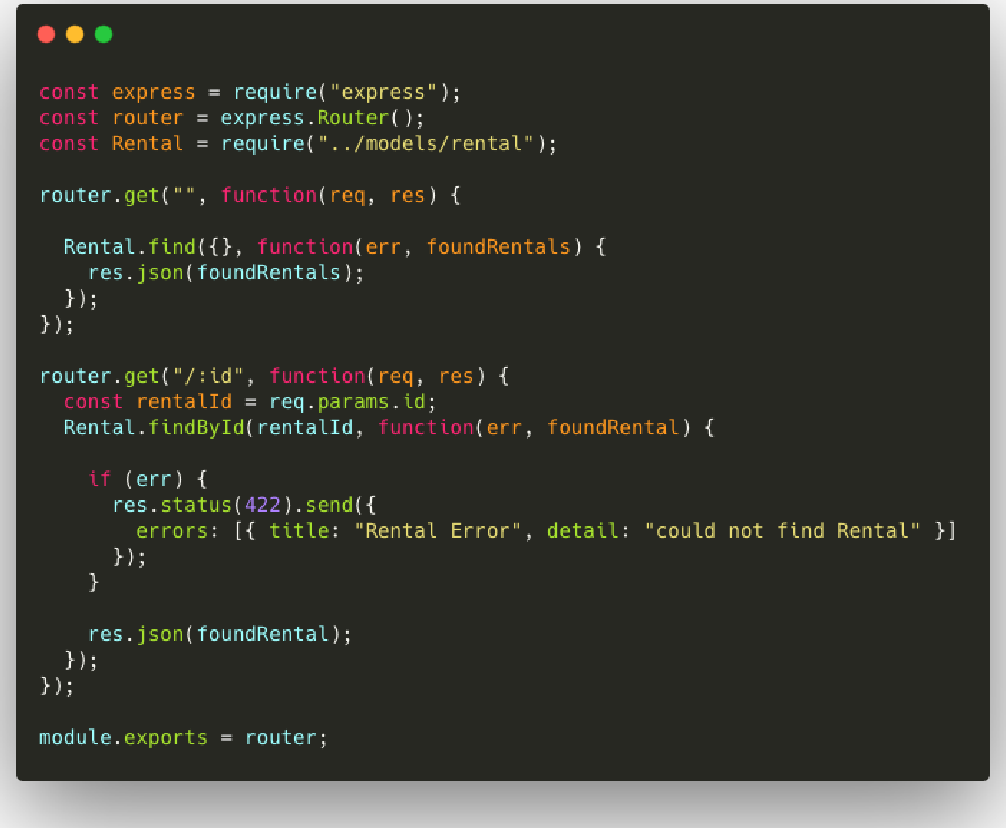
```
{
  "_id": {
    "$oid": "5e10c0bd6b8be82533863426"
  },
  "title": "Family Apartment in Oulu",
  "city": "oulu",
  "street": "Kotkantie 1, 04320 Tuusula",
  "category": "house",
  "image": "https://s.iha.com/592400008414/Short-term-rentals-Lans-Haus-Olympia_8.jpeg",
  "bedrooms": {
    "$numberInt": "3"
  },
  "description": "80 square metere apartment with private sauna is available for rent",
  "dailyRate": {
    "$numberInt": "930"
  },
  "shared": true,
  "createdAt": {
    "$date": {
      "$numberLong": "1578156221697"
    }
  },
  "__v": {
    "$numberInt": "0"
  }
}
```

*FIGURE 28. Rental Data saved in the JSON format in MongoDB*

### 5.3.4 Restful API

API (Application Programming Interface) is an interface which allows one application to communicate with other services through the Internet following some set of rules[18]. An existing API can be used as a part of the application which facilitates the services that would otherwise be very hard for a developer to crack. For example, a developer can stretch the functionality of the website just by integrating the third party free to use an open API such as weather API, News API, Cryptocurrency tracker API, Currency Exchange API and Movies API. These APIs help to gather information and create a content which would otherwise be very unlikely for a developer to access.

REST stands for the Representational State Transfer. It is a software architecture for creating web services. Certain principles must be followed for an application to be called RESTful. It uses HTTP requests to GET, PUT, POST and DELETE data[19]. Figure 29 below shows the in-app API to get rental.



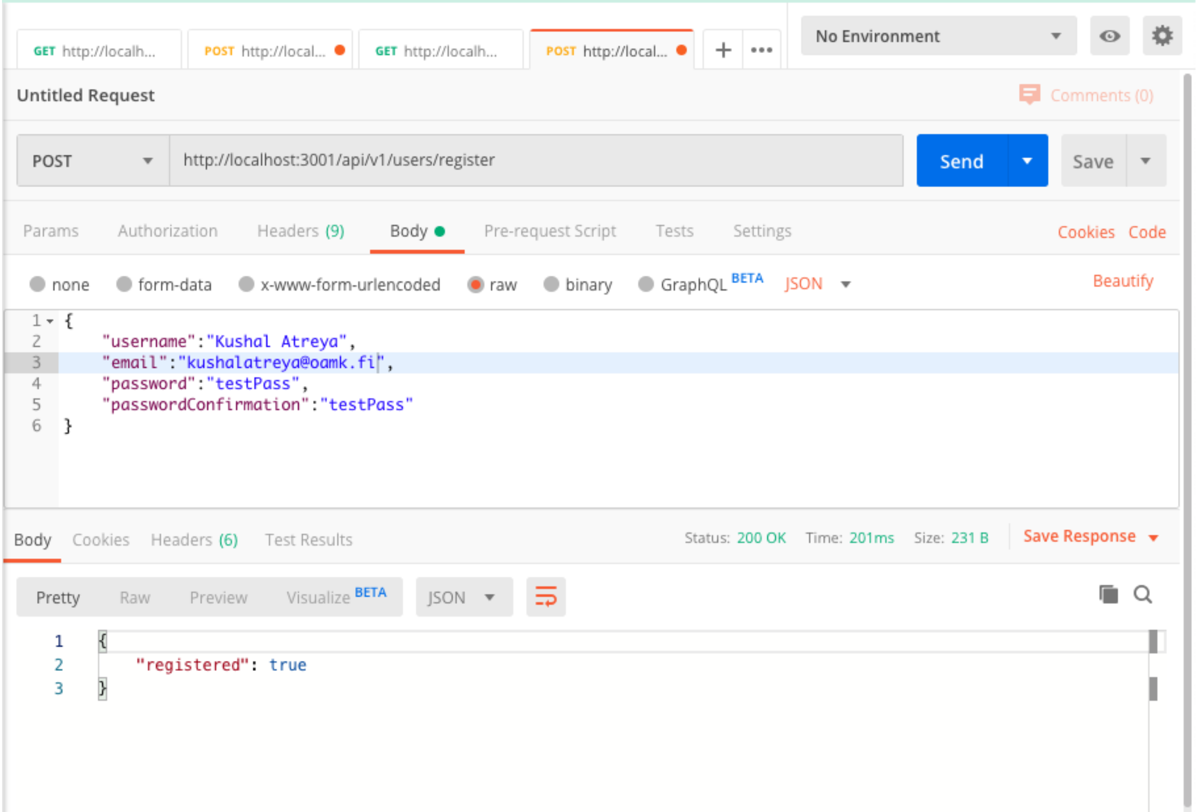
```
const express = require("express");
const router = express.Router();
const Rental = require("../models/rental");

router.get("", function(req, res) {

  Rental.find({}, function(err, foundRentals) {
    res.json(foundRentals);
  });
});

router.get("/:id", function(req, res) {
  const rentalId = req.params.id;
  Rental.findById(rentalId, function(err, foundRental) {

    if (err) {
      res.status(422).send({
        errors: [{ title: "Rental Error", detail: "could not find Rental" }]
      });
    }

    res.json(foundRental);
  });
});

module.exports = router;
```

*FIGURE 29. API to get rental.*

**PostMan:**

A PostMan is a tool, which eases the building or working with APIs. It is used for checking the API response, creating API specifications, debugging, running test automation, monitoring, documenting and publishing APIs[20] . The authorization and header can be set if working with the user validation or the user group. Such a test can be created and finalized before completing the front-end section or waiting for the user interaction. It is also used to simulate endpoints and responses without building a real server.

PostMan is free and easy to start and it has a great community and support[21]. It was used in this application to test if APIs were working correctly and in multiple environments until everything was predictable.



POST http://localhost:3001/api/v1/users/register

Body: ● raw JSON

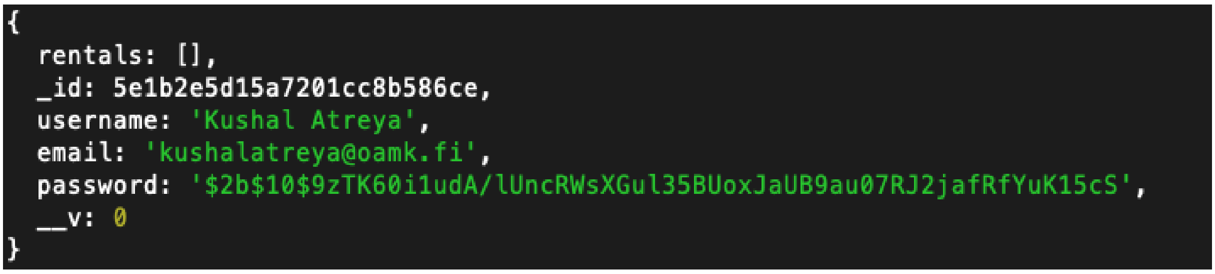
```
{
    "username":"Kushal Atreya",
    "email":"kushalatreya@oamk.fi",
    "password":"testPass",
    "passwordConfirmation":"testPass"
}
```

Status: 200 OK Time: 201ms Size: 231 B

JSON

```
{
    "registered": true
}
```

*FIGURE 30. Using PostMan to create the user-data prior of having the front-end login system.*



```
{
  rentals: [],
  _id: 5e1b2e5d15a7201cc8b586ce,
  username: 'Kushal Atreya',
  email: 'kushalatreya@oamk.fi',
  password: '$2b$10$9zTK60i1udA/lUncRWsXGul35BUoxJaUB9au07RJ2jafRfYuK15cS',
  __v: 0
}
```

*FIGURE 31. Response of created user with encrypted password from the console.*

Figure 30 above showcases the use of Post-Man to create user credentials without the front-end login system and figure 31 shows the response.

### 5.4 Working Methodology

Identification of the problem and turning it to a business model to provide a reliable solution is in itself a tricky part. It involves acute observation, intense survey and behavioural studies. Furthermore, a market analysis, a competitor analysis and a focused target group should be identified. One can put all the resources and effort and energy, but new elements always pop up and collectively adds more sense to the project. This project turned out to be huge. Many additional extensions and branches could be added to make this project more sensible and stronger. Due to limitation of time and resources and the scope of the thesis, few essential parts were only focused and taken into effect. After which, user-stories were designed following with the wireframe and mock-ups.

The design was initiated with keeping the user interaction and needs in mind, visualizing the purpose of the project and target group and their views. Balsamiq was used for creating the wireframe, but it was later opted out for Figma. Figma, being modern and visually appealing, was very intuitive and productive. The Front-end project was developed using create-react-app, Bootstraps for responsive design and Redux as a state management tool. Google Map API was used for map services and to show the location of the rentals. Express was used in the backend alongside with middleware, encryption, JSON web tokens and validation. Many NPM packages and APIs were integrated to develop web-services, stretching the functionalities of the application such as Google Maps API, Bycript for salting and encrypting user password.

Trello was used for keeping track of the plan and for visualizing the entire playing-field of the application with the acknowledgement of what is coming next. Git was used as a version control.

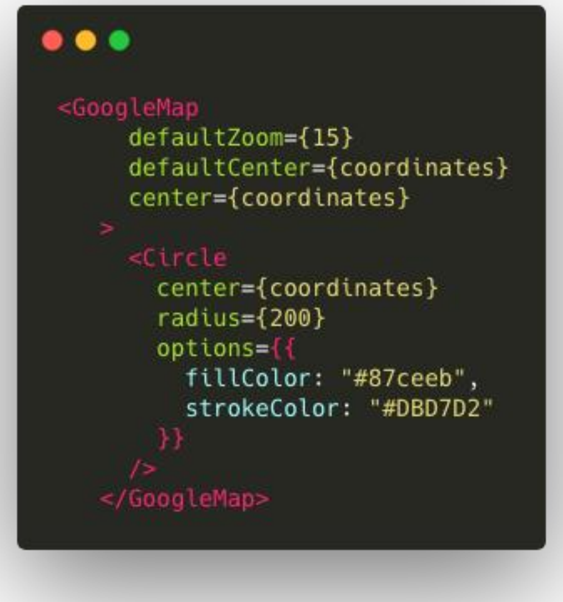
#### 5.4.1 Challenges and learning curves

The web app was a full-fledged full-stack project using the modern technology. Many a time, there were technical issues, short of knowledge and necessity to learn new skills. Relying on many small libraries, frameworks, modules and trying to integrate everything to build something out of them did not always work as intended.

The project was initiated with ‘create-react-app’ using the NPM. All the required packages were added in the node-modules, some of which created a version conflict. The version conflict was fixed by auditing the NPM packages one by one. There were problems while importing modules of react-redux, redux-thunk and react-router-dom. This was fixed by fixing the relative paths, downloading the latest version and by adding an additional flag and suffix to it, for example, `npm install --save react-router-dom@latest` or by mentioning the desired version of the package like: `npm install --save react-redux@5.0.7 redux@3.7.2 redux-thunk@2.2.0`. There were many errors and sort-backs on the way. Most of the errors and un-intended output were managed by exception handling, setting the error message and debugging the code.

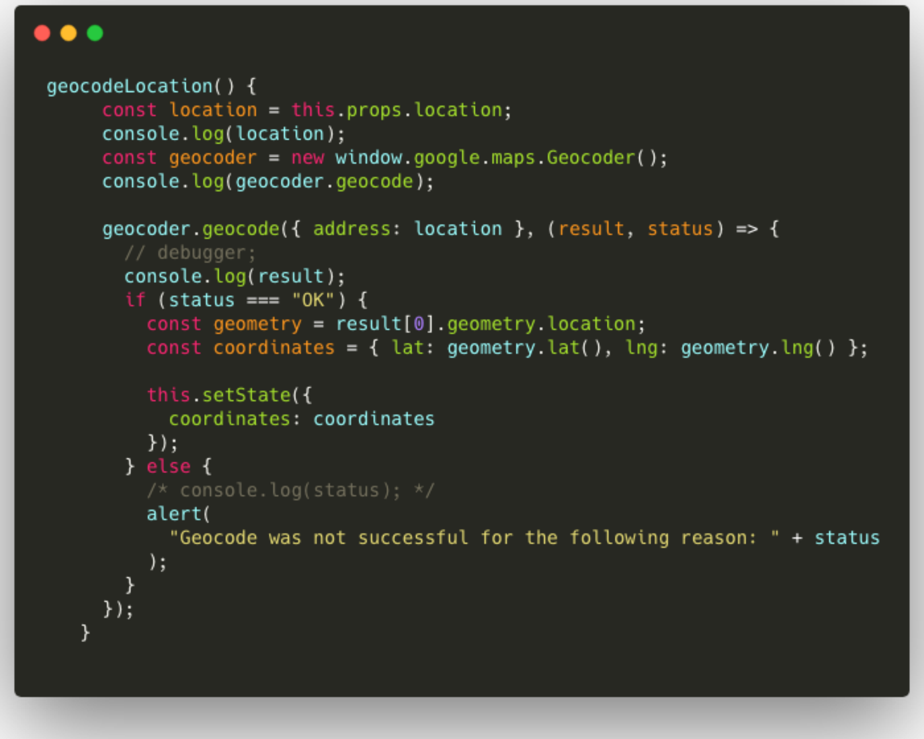
With the database, there were many problems because of lack of experience with the tools. Many `‘null’` and `‘undefined’` responses were result of using wrong relative paths and weak schema. Fetching the data from the database was performed using JavaScript iteration methods. Furthermore, there existed deprecation warnings while trying to pass multiple arguments in MongoDB. This was fixed by adding `{ useUnifiedTopology: true, useNewUrlParser: true }` to the code.

Pinpointing the location of the rental property was an essential feature in the project. The exact location was not to be displayed because of the security feature and business perspective. The radius of the location was preferred to solve this problem by replacing the marker as shown in the figure 32. Updating the latitude and longitude of the rental property and manually updating the rental location was not the scalable idea. Google Map’s Geocoding script was used as in figure 33, which shows the location from the address provided by the landlord. Besides this, there were error messages and watermarks shown on the displayed map, which was resolved by making a developer account and enabling Geolocation API and Geocoding API and opening the billing account in the Google Cloud Platform.



```
<GoogleMap
     defaultZoom={15}
     defaultCenter={coordinates}
     center={coordinates}
   >
     <Circle
       center={coordinates}
       radius={200}
       options={{
         fillColor: "#87ceeb",
         strokeColor: "#DBD7D2"
       }}
     />
   </GoogleMap>
```

*Figure 32. Using circle and radius of 200 units, to showcase the location of the rental.*



```
geocodeLocation() {
     const location = this.props.location;
     console.log(location);
     const geocoder = new window.google.maps.Geocoder();
     console.log(geocoder.geocode);

     geocoder.geocode({ address: location }, (result, status) => {
       // debugger;
       console.log(result);
       if (status === "OK") {
         const geometry = result[0].geometry.location;
         const coordinates = { lat: geometry.lat(), lng: geometry.lng() };

         this.setState({
           coordinates: coordinates
         });
       } else {
         /* console.log(status); */
         alert(
           "Geocode was not successful for the following reason: " + status
         );
       }
     });
   }
```

*Figure 33. Geocoding using the location provided in the rentals.*

# 6 CONCLUSION AND DISCUSSION

During the journey of writing the thesis and developing the project, the author has gained immense knowledge in the related field of survey, interview, observation, extensive research and analysis of the outcome and funnelling it to extract reasonable findings. Besides this, interpersonal skills, such as active listening, leadership, patience, dependability, teamwork, decision making, confidentiality and communication skills, were improving with time. Working in the project as a developer not only contributed to programming skills but also groomed as a better problem-solver, enhanced the ability to work under pressure and most importantly brought the insight that things can be done or built without having prior knowledge about the tools to be used in the project. Such phycological impact, interpersonal skills and field knowledge will surely make the author a better person in personal life and experienced and competent personality in professional life ahead.

The web project was finalised in the form of a working prototype which demonstrates the user interface and on how things can be carried on further. The prototype is functional and serves the purpose of the thesis. Due to the limitation of time and resources and minimisation of the scope of the thesis, everything which could be a full-fledged large-scale and working web-platform was not created.

Moreover, the development and search for the potential investors are in the process as launching the application for the large-scale market necessitates more developing hands, forces from diverse fields, capital for maintaining the application, a database, a cloud storage, advertisements, premium support and marketing.

# REFERENCES

1. B. Tracey. 2017. Date of Retrieval Jan 1, 2020 https://www.gsmasia.net/3g-bands.html

2. Bhusal, N. 2019. 5 Best Digital Wallets in Nepal. TechSathi .Date of Retrieval Jan 1, 2020. https://techsathi.com/best-digital-wallets-in-nepal/

3. Anonymous. 2017. One in two Nepalis use internet: NTA report. Date of Retrieval Jan 1, 2020 https://kathmandupost.com/money/2017/05/09/one-in-two-nepalis-use-internet-nta-report

4. Anonymous 2016. 42 percent population lives in urban centres. Date of Retrieval Mar 23, 2020 http://kathmandupost.ekantipur.com.np/printedition/news/2016-05-28/42-percent-population-lives-in-urban-centres.html

5. Central Bureau of Statistics. 2012. 2011 Census. Highlights of the census results. National Population and Housing Census 2011. Date of retrieval Jan 1, 2020 https://unstats.un.org/unsd/demographic-social/census/documents/Nepal/Nepal-Census-2011-Vol1.pdf

6. Ilya, K. 2019. Code editors. Date of Retrieval Jan 9, 2020 https://javascript.info/code-editors

7. Anonymous. 2020. Documentation for Visual Studio Code. Date of Retrieval Jan 9, 2020 https://code.visualstudio.com/docsz

8. Anonymous. 2019. Chrome DevTools | Tools for Web Developers. Date of Retrieval Jan 11, 2020 https://developers.google.com/web/tools/chrome-devtools

9. Kadivar, N. 2018. Top 10 Version Control Systems. Date of Retrieval Jan 1, 2020 https://hackernoon.com/top-10-version-control-systems-4d314cf7adea

10. CCI Learning. 2017. IC3 Certification Guide Using Microsoft Windows 10 & Microsoft Office 2016. 2nd Edition, 420

11. contributors, Mark Otto, Jacob Thornton, and Bootstrap. 2019. Bootstrap. Date of Retrieval Jan 11, 2020 https://getbootstrap.com/

12. Ranger, S. Programming languages: JavaScript most used, Python most studied, Go most promising. Date of Retrieval Jan 11, 2020 https://www.zdnet.com/article/programming-languages-javascript-most-used-python-most-studied-go-most-promising/

13. Anonymous. 2019. React – A JavaScript library for building user interfaces. Date of Retrieval Jan 12, 2020 https://reactjs.org/

14. Redux documentation authors. Redux - A predictable state container for JavaScript apps. | Redux. Date of Retrieval Jan 12, 2020 https://redux.js.org/

15. Anonymous. 2020. Node.js – Introduction. Date of Retrieval Jan 2, 2020 https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm

16. MDN contributors. 2020. Express/Node introduction. Date of Retrieval Jan 24, 2020 https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction

17. Ahlawat, A. 2019. Introduction to MongoDB | MongoDB Tutorial | Studytonight. Date of Retrieval Jan 24, 2020 https://www.studytonight.com/mongodb/introduction-to-mongodb

18. Anonymous. What is an API? (Application Programming Interface). Date of Retrieval Jan 30, 2020
https://www.mulesoft.com/resources/api/what-is-an-api

19. Rouse, M. 2019. What is a RESTful API (REST API) and How Does it Work? Date of Retrieval Jan 30, 2020
https://searchapparchitecture.techtarget.com/definition/RESTful-API

20. Postman documentation authors. 2019. Postman | Tools. Date of Retrieval Jan 30, 2020
https://www.postman.com

# APPENDICES

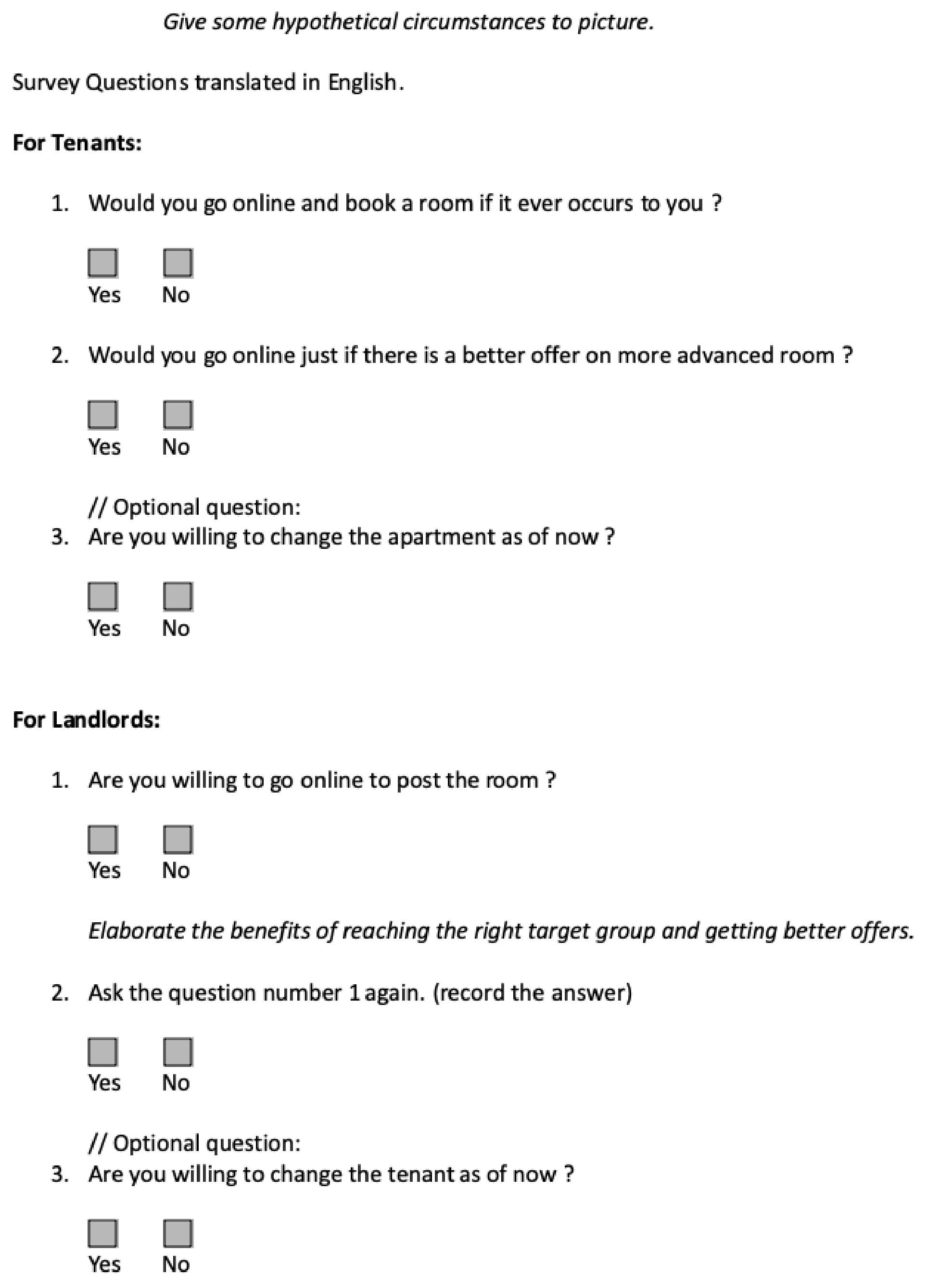
List of Appendices:

Appendix 1: Survey questions

Appendix 2: UML diagram

# APPENDIX 1

Survey questions



*Give some hypothetical circumstances to picture.*

Survey Questions translated in English.

**For Tenants:**

1. Would you go online and book a room if it ever occurs to you ?

   □ Yes □ No

2. Would you go online just if there is a better offer on more advanced room ?

   □ Yes □ No

   // Optional question:

3. Are you willing to change the apartment as of now ?

   □ Yes □ No

**For Landlords:**

1. Are you willing to go online to post the room ?

   □ Yes □ No

   *Elaborate the benefits of reaching the right target group and getting better offers.*

2. Ask the question number 1 again. (record the answer)

   □ Yes □ No

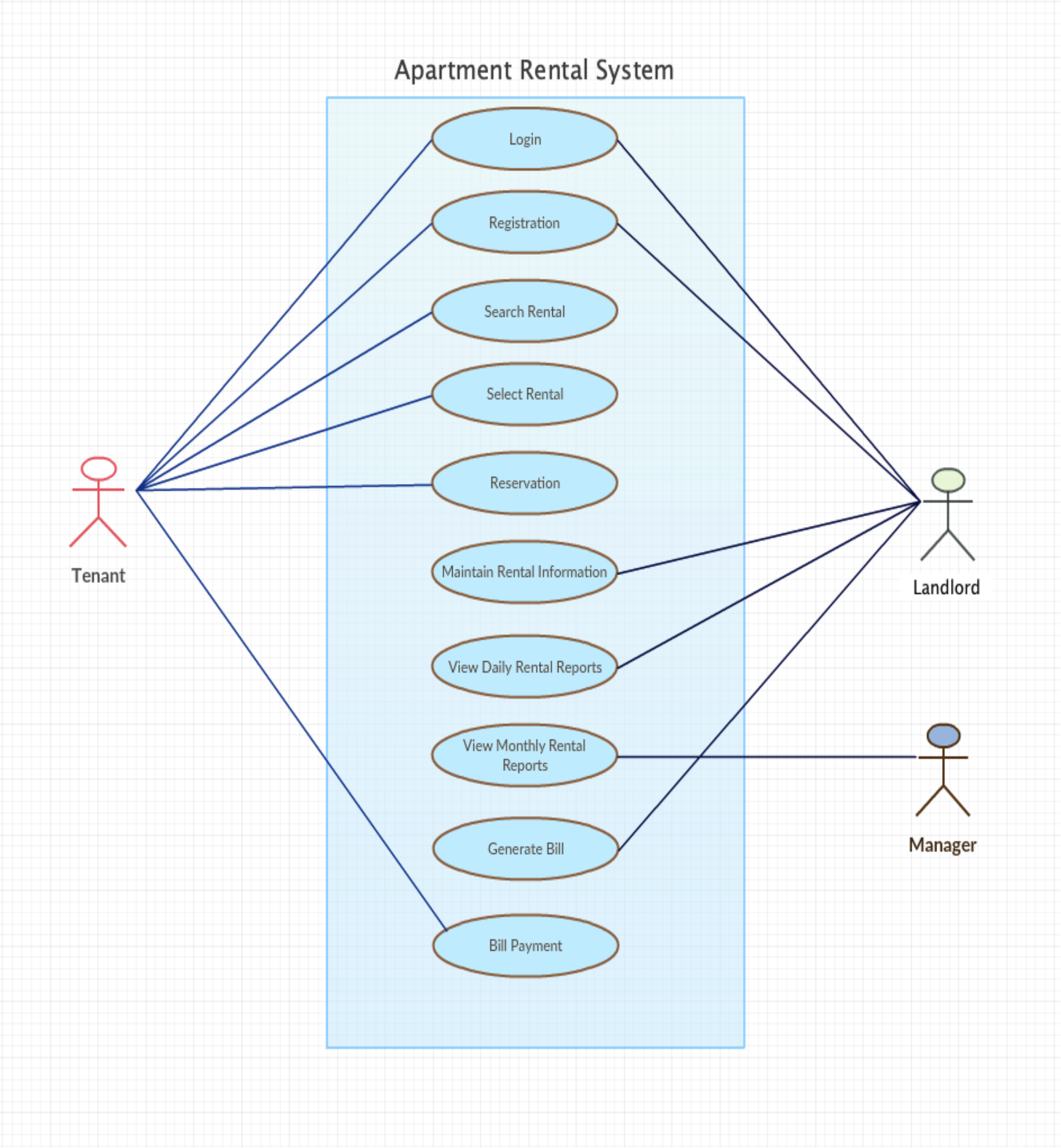
   // Optional question:

3. Are you willing to change the tenant as of now ?

   □ Yes □ No

# APPENDIX 2

UML diagram


Apartment Rental System
Login
Registration
Search Rental
Select Rental
Reservation
Maintain Rental Information
View Daily Rental Reports
View Monthly Rental Reports
Generate Bill
Bill Payment
Tenant
Landlord
Manager