HOME RENTAL APP

A PROJECT REPORT

Submitted for the partial fulfillment of the requirement for the degree of

MASTER OF SCIENCE
IN
COMPUTER SCIENCE

Submitted by

A. MOHAMMED SHAFEEQUE
2213182078017

Under the Guidance of

Dr. G. NAJEEB AHMED M.Sc., M.Phil., Ph.D.

P.G DEPARTMENT OF COMPUTER SCIENCE

THE NEW COLLEGE

CHENNAI – 600 014

APRIL – 2024

BONAFIDE CERTIFICATE

This is to certify that the entitled "**HOME RENTAL APP"** being submitted to the University of Madras, Chennai by **A. MOHAMMED SHAFEEQUE** Reg.No:2213182078017 for the partial fulfilment for the award of degree of Master of Science in Computer Science is a bonafide record of work carried out by him under my guidance and supervision.

Dr. G Najeeb Ahmed
Project Guide

Dr. P Hakkim Divan Mydeen
Head of the Department

Submitted for the project Viva Voice Examination in THE NEW COLLEGE, Chennai held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

Date:

Place:

ACKNOWLEDGEMENT

First of all, I thank the Almighty for blessing me with his abundance grace in completing my project successfully.

I express my sincere gratitude to the Principal **Dr. M. Asrar Sheriff M.Sc., M.Phil., Ph.D.,** for permitting me to do the project with fullest spirit.

I am pleased to acknowledge my great thanks to **Dr. P. Hakkim Divan Mydeen M.Sc., M.Phil., Ph.D.,** Head of the Department of Computer Science, for his useful guidance and encouragement to complete this project.

My Sincere thanks to my guide **Dr. G. Najeeb Ahmed M.Sc., M.Phil., Ph.D.,** Department of Computer Science, for his valuable guidance and encouragement for finishing this project successfully.

I thank all the Staff Members of Computer Science for their co-operation in completing this project.

I acknowledge my heartfelt thanks to my parents for their encouragement, social and economic support for completing this project successfully.

A. MOHAMMED SHAFEEQUE

TABLE OF CONTENTS

INTRODUCTION

ABOUT THE PROJECT

In the digital age, the demand for efficient platforms facilitating various aspects of daily life has surged. Particularly, the need for robust and user-friendly home rental websites has become paramount. Our project endeavors to meet this demand by conceptualizing and implementing a comprehensive home rental platform.

This project represents the culmination of our academic journey, combining theoretical knowledge with practical application. Through meticulous planning and iterative development, we aim to create a dynamic and user-centric platform that addresses the diverse needs of renters and landlords alike.

This documentation outlines the architecture, design principles, functionalities, and implementation details of our home rental website. It encompasses key components such as the login page, registration page, home listings, user profiles, messaging system, and more.

Our primary goal is to deliver a platform that not only facilitates seamless property search and rental transactions but also prioritizes user privacy and security. By leveraging modern web technologies and adhering to best practices, we aspire to create a scalable and resilient solution capable of accommodating the evolving needs of our users.

This documentation serves as a guide for understanding the intricacies of our home rental platform, providing insights into design rationale, implementation strategies, and potential avenues for future expansion. We aspire for this project to not only fulfill academic requirements but also contribute to the ongoing dialogue on real estate technology and its societal impact.

Login Page:

Users can access their accounts using their unique username and password, allowing them to manage their rental listings or search for properties.

Sign Up:

New users can register by providing basic details such as name, email, username, and password, enabling them to create a profile and start listing properties or searching for rentals.

Home Page:

The home page displays all available rental properties, along with personalized recommendations based on user preferences. Users can switch between light and dark modes for a customized browsing experience.

My Profile:

Users can view and manage their rental listings, preferences, and account details from this section.

Property Profile:

This page provides detailed information about a selected rental property, including photos, amenities, pricing, and contact information for the landlord.

Messaging:

The messaging section allows users to communicate with landlords or potential tenants, facilitating seamless interaction and negotiation.

SYSTEM ANALYSIS

Existing System vs. Proposed System:

Target Audience and Purpose:

While existing rental platforms may cater primarily to specific demographics, our home rental application targets a broader audience, including tenants, landlords, and property managers. It aims to streamline the rental process for individuals across various lifestyles and preferences.

Content Diversity and Accessibility:

Unlike traditional rental platforms that may offer limited property types or locations, our application embraces diversity by featuring a wide range of rental properties, including apartments, houses, and commercial spaces. It ensures accessibility by providing an intuitive interface for users to explore and interact with listings effortlessly.

Community Engagement and Networking:

Our platform goes beyond basic listing functionalities by fostering community engagement among renters and landlords. Users can join interest-based groups, share rental experiences, and seek advice from peers, creating a vibrant ecosystem of collaboration and support.

Personalization and User Experience:

With a focus on user-centric design, our application offers personalized recommendations and customizable features to enhance the rental experience. From tailored search filters to interactive property tours, we prioritize user satisfaction and convenience at every step.

REQUIREMENT SPECIFICATION

Hardware Requirements:

Processor: Intel Pentium-IV

RAM: 2.00 GB

Monitor: Standard LCD/LED Monitor

Hard Disk: 300GB

Keyboard: Standard Keyboard

Mouse: Standard Mouse

Software Requirements:

Front End: React (HTML, CSS, JS)

Back End: Express (Node.js)

Database: MongoDB

Operating System: Windows 7

SYSTEM SPECIFICATION

Introduction:

The process of designing our home rental application involves defining its architecture, components, modules, interfaces, and data to meet specific requirements. This includes applying system theory to product development and configuring the system to fulfill identified needs. System design encompasses both high-level architecture and detailed design, ensuring the effective operation of individual components within the overall system.

Input Design:

Input design plays a crucial role in ensuring the quality of system output. It involves specifying how data is entered into the system and processed. Effective input design ensures the reliability of the system and the production of accurate results from user-originated inputs. The objectives of input design include effectiveness, accuracy, ease of use, consistency, and attractiveness. In our home rental application, input design focuses on converting user inputs into a format compatible with mobile devices. We utilize various tools such as text boxes, labels, list boxes, radio groups, buttons, and image views to facilitate easy data entry. Proper validation and control of input data reduce the chance of errors, ensuring a reliable and user-friendly experience for our users.

Output Design:

Output design is essential for delivering information to users effectively. The output of our home rental application includes property listings, search results, and messaging interfaces. It is designed to be attractive, convenient, and informative, enhancing the user experience and facilitating decision-making. Forms are designed with various features to make the output visually pleasing and easy to understand. By prioritizing the design of outputs, we aim to improve the system's relationship with users and enhance its usefulness. Form design elaborates on how output is presented and the layout available for capturing information, ensuring that users can easily access and interpret the information provided by the application.

SOFTWARE DESCRIPTION

ABOUT THE FRONT END:

Front End Development:

In our home rental application, the front end refers to the part of the platform that users directly interact with, also known as the "client-side." This includes everything users experience directly, such as text, images, buttons, navigation menus, and more. Front end development is implemented using HTML, CSS, and JavaScript (React), which are essential languages for building user interfaces on websites, web applications, and mobile apps. Front end developers are responsible for designing the structure, appearance, behavior, and content of all elements visible on browser screens.

HTML (Hypertext Markup Language):

HTML is used to design the front-end portion of web pages using a markup language. It defines the structure and layout of web content, including text, images, links, and other elements. In our home rental application, HTML is utilized to create the foundational structure of the user interface, ensuring proper organization and accessibility of content.

CSS (Cascading Style Sheets):

CSS is a language designed to simplify the process of making web pages presentable by applying styles to HTML elements. It allows developers to define the visual appearance of elements, including colors, fonts, layouts, and animations.

In our application, CSS is used to enhance the presentation of user interface elements, ensuring a visually appealing and consistent design across all pages.

JavaScript:

JavaScript is a scripting language used to enhance the functionality of websites by making them interactive and dynamic. It enables developers to create responsive features, handle user input, and manipulate web page content in real-time. In our home rental application, JavaScript is utilized to implement interactive elements, validate user inputs, and facilitate smooth user interactions, enhancing the overall user experience.

React:

React is a free and open-source front-end JavaScript library used for building user interfaces based on reusable components. It simplifies the process of developing complex user interfaces by breaking them down into smaller, reusable components.

In our application, React is employed to create modular and efficient UI components, allowing for easier maintenance, scalability, and code reuse.

Back End Technologies:

Node.js:

Node.js is a cross-platform, open-source JavaScript runtime environment that allows developers to execute JavaScript code outside of a web browser. It provides a scalable and efficient platform for building server-side applications and APIs. In our home rental application, Node.js is utilized for server-side scripting, enabling

the handling of HTTP requests, data processing, and business logic implementation.

Express.js:

Express.js, or simply Express, is a back end web application framework for building RESTful APIs with Node.js. It provides a robust set of features for routing, middleware, and HTTP utilities, making it ideal for developing web applications and APIs. In our application, Express.js is used to create RESTful endpoints for handling various operations, such as user authentication, property listing, and messaging.

Database Management:

MongoDB:

MongoDB is an open-source document-oriented database designed to store large-scale data efficiently. It belongs to the NoSQL (Not only SQL) database category, offering flexible storage and retrieval options for unstructured data. In our home rental application, MongoDB is employed as the primary database for storing property listings, user profiles, and messaging data. It provides official driver support for various programming languages, ensuring compatibility and ease of integration with our application's backend architecture. Additionally, MongoDB offers scalability and performance benefits, making it suitable for managing the diverse data requirements of our platform.

SYSTEM TESTING & IMPLEMENTS

System Testing and Implementation:

Implementation involves carrying out, executing, or practicing a plan, method, or design for accomplishing a task. In the context of information technology, implementation encompasses all processes required to ensure new software or hardware operates properly in its environment. This includes installation, configuration, running, testing, and making necessary changes.

Testing:

Testing is the process of executing a program or system with the intent of finding errors or evaluating its attributes and capabilities to ensure it meets required results. Unlike physical systems that fail in predictable ways, software can fail in various unexpected ways, making it challenging to detect all failure modes.

Unit Testing:

Unit testing is a software development process where the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. It focuses on testing modules and routines that perform specific functions to locate errors. In our home rental application, unit testing is applied to the registration form, ensuring that fields are validated under various conditions. For example, if fields are left empty or contain invalid data, the browser displays alert messages to prompt users for correct inputs.

Integration Testing:

Integration testing, also known as Integration and Testing (I&T), involves combining and testing program units as groups in multiple ways. Two major methods of integration testing include bottom-up and top-down approaches. Bottom-up testing begins with unit testing, followed by progressively higher-level combinations of units. In contrast, top-down testing starts with testing higher-level modules first and then progressively tests lower-level modules. In our application, integration testing is applied after attaching all Node.js programs to appropriate links, verifying the functionality of the system through interaction with various components.

Acceptance Testing:

Acceptance testing is conducted to verify that the system meets specified requirements and performs as expected. It ensures that file sizes are adequate, indexes are built properly, and sorting and rendering procedures function correctly at the system level. In our home rental application, acceptance testing ensures that the platform meets user expectations regarding functionality, performance, and reliability, providing a seamless experience for tenants and landlords alike.

ER DIAGRAM

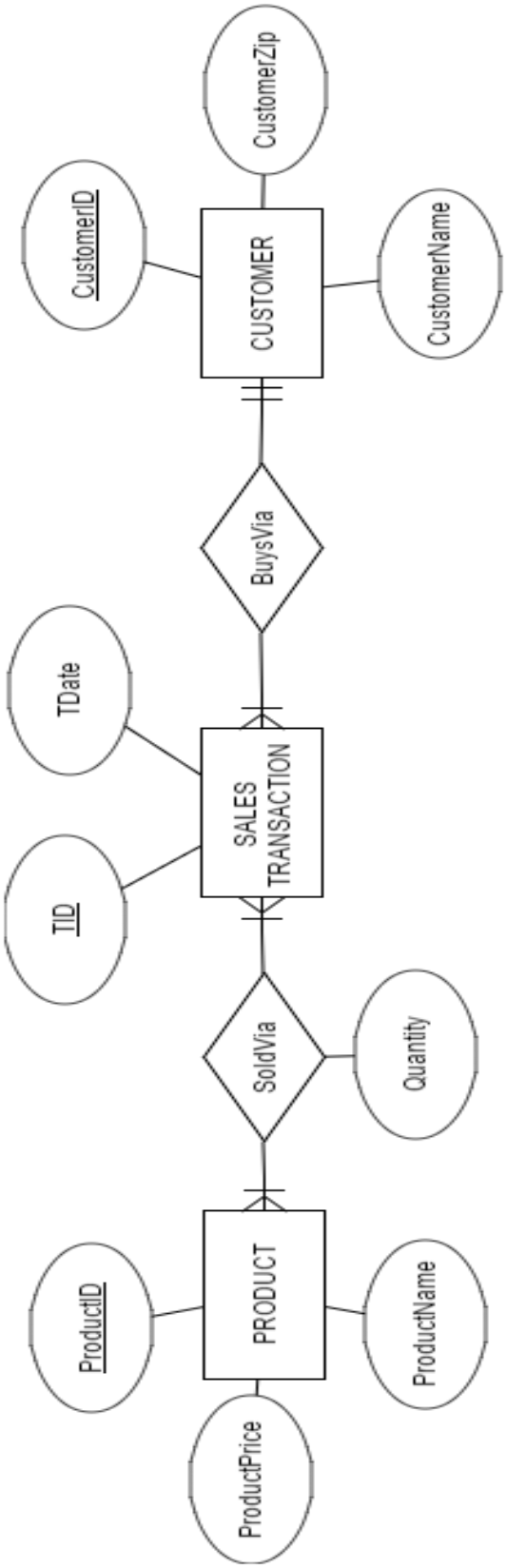
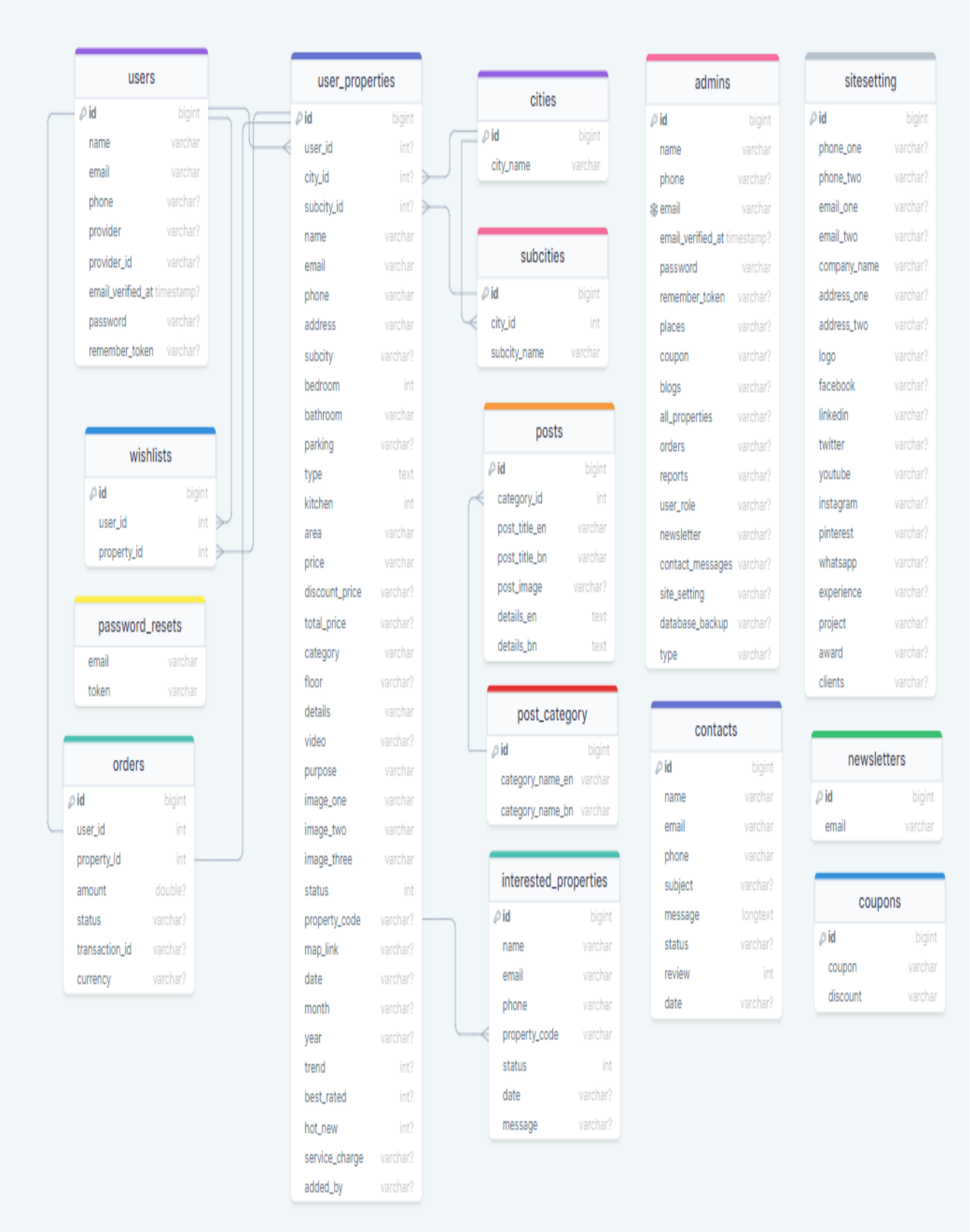
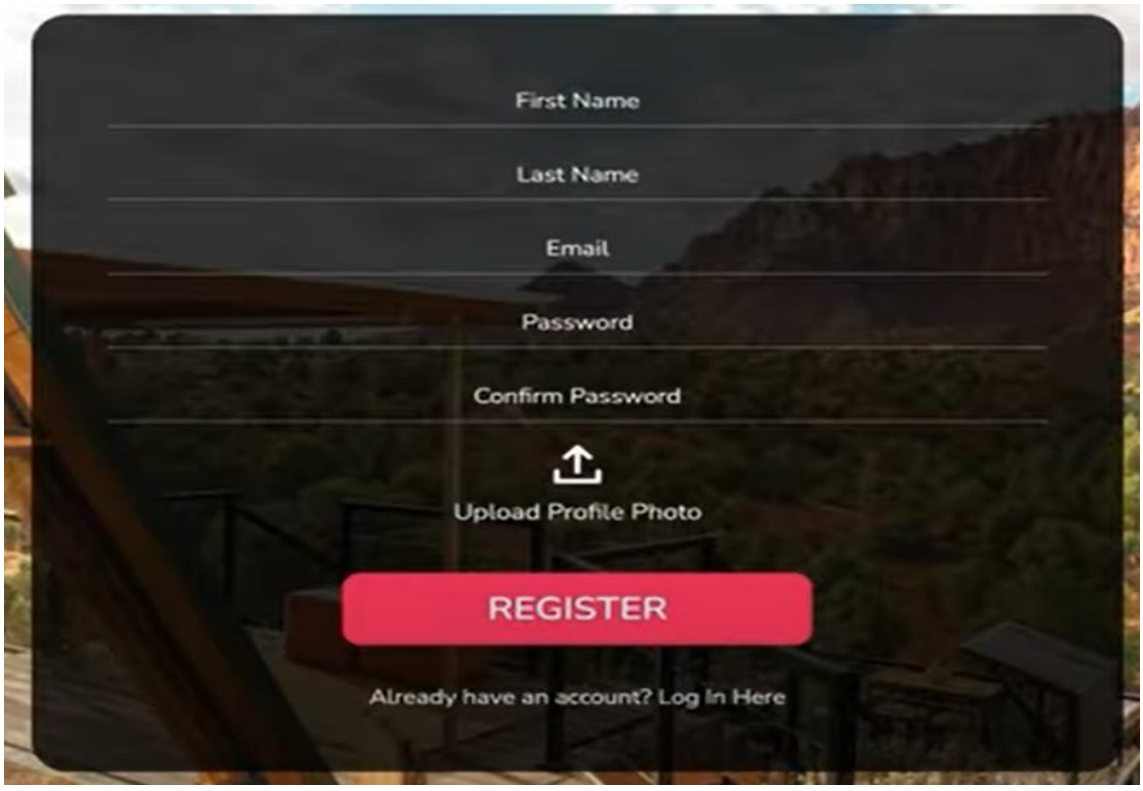
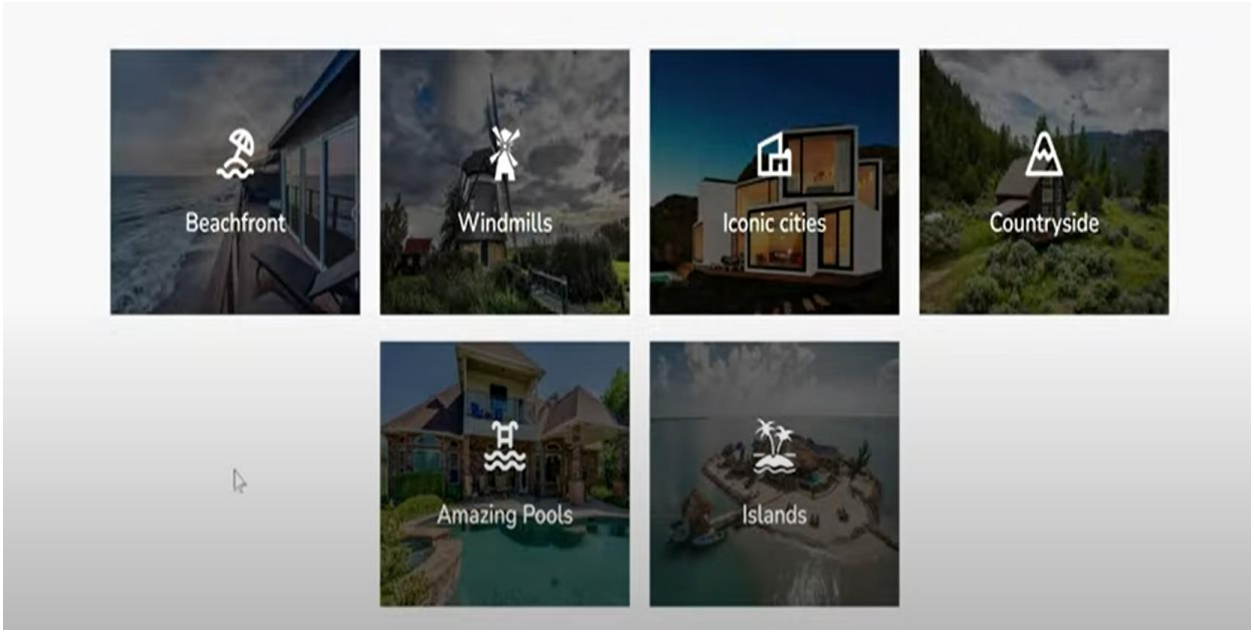

CustomerZip
CustomerID
CUSTOMER
CustomerName
BuysVia
TDate
SALES TRANSACTION
TID
SoldVia
Quantity
ProductID
PRODUCT
ProductName
ProductPrice

TABLE STRUCTURE

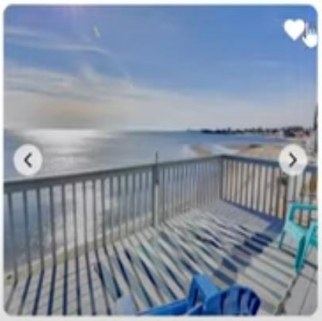

users
id bigint
name varchar
email varchar
phone varchar?
provider varchar?
provider_id varchar?
email_verified_at timestamp?
password varchar?
remember_token varchar?
user_properties
id bigint
user_id int?
city_id int?
subcity_id int?
name varchar
email varchar
phone varchar
address varchar
subcity varchar?
bedroom int
bathroom varchar
parking varchar?
type text
kitchen int
area varchar
price varchar
discount_price varchar?
total_price varchar?
category varchar
floor varchar?
details varchar
video varchar?
purpose varchar
image_one varchar
image_two varchar
image_three varchar
status int
property_code varchar?
map_link varchar?
date varchar?
month varchar?
year varchar?
trend int?
best_rated int?
hot_new int?
service_charge varchar?
added_by varchar?
cities
id bigint
city_name varchar
subcities
id bigint
city_id int
subcity_name varchar
admins
id bigint
name varchar
phone varchar?
email varchar
email_verified_at timestamp?
password varchar
remember_token varchar?
places varchar?
coupon varchar?
blogs varchar?
all_properties varchar?
orders varchar?
reports varchar?
user_role varchar?
newsletter varchar?
contact_messages varchar?
site_setting varchar?
database_backup varchar?
type varchar?
sitesetting
id bigint
phone_one varchar?
phone_two varchar?
email_one varchar?
email_two varchar?
company_name varchar?
address_one varchar?
address_two varchar?
logo varchar?
facebook varchar?
linkedin varchar?
twitter varchar?
youtube varchar?
instagram varchar?
pinterest varchar?
whatsapp varchar?
experience varchar?
project varchar?
award varchar?
clients varchar?
wishlists
id bigint
user_id int
property_id int
password_resets
email varchar
token varchar
posts
id bigint
category_id int
post_title_en varchar
post_title_bn varchar
post_image varchar?
details_en text
details_bn text
post_category
id bigint
category_name_en varchar
category_name_bn varchar
orders
id bigint
user_id int
property_id int
amount double?
status varchar?
transaction_id varchar?
currency varchar?
interested_properties
id bigint
name varchar
email varchar
phone varchar
property_code varchar
status int
date varchar?
message varchar?
contacts
id bigint
name varchar
email varchar
phone varchar
subject varchar?
message longtext
status varchar?
review int
date varchar?
newsletters
id bigint
email varchar
coupons
id bigint
coupon varchar
discount varchar

DESIGN LAYOUT

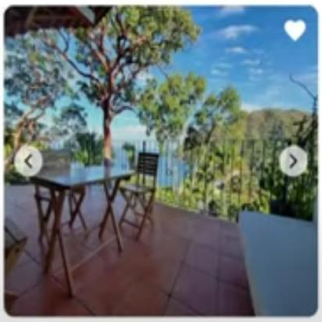

First Name
Last Name
Email
Password
Confirm Password
Upload Profile Photo
REGISTER
Already have an account? Log In Here


Beachfront
Windmills
Iconic cities
Countryside
Amazing Pools
Islands

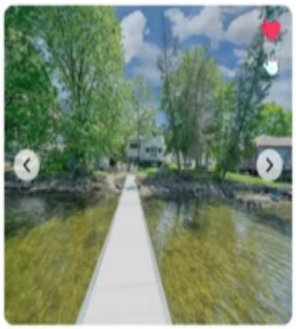
Beachfront



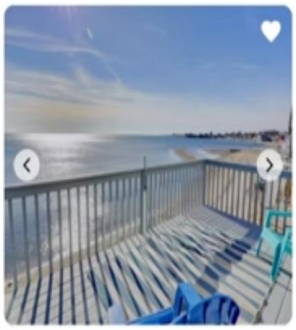
Milford, Connecticut, United States
Beachfront
An entire place
$713 per night



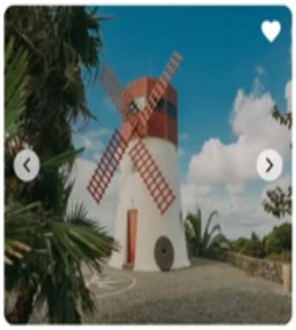
Yelapa, Jal, Mexico
Beachfront
A Shared Room
$176 per night



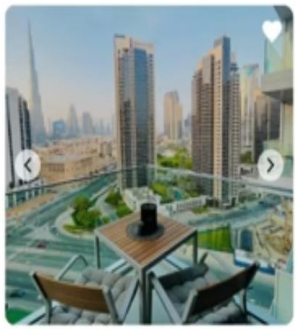
Barrie, Ontario, Canada
Lakefront
An entire place
$319 per night



Milford, Connecticut, United States
Beachfront
An entire place
$713 per night



Ponta Delgada, Ilhas, Portugal
Windmills
An entire place
$228 per night



Dubai, Dubai, Arab
Iconic cities
An entire place
$170 per night

What this place offers?

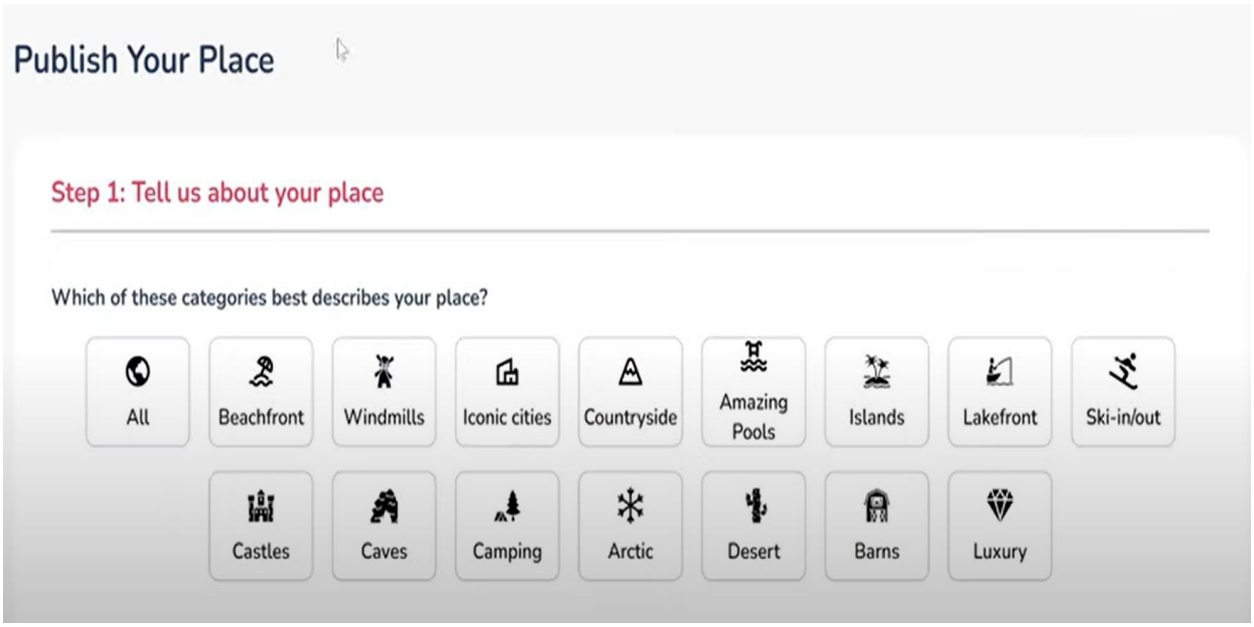
- Outdoor shower
- Bath tub
- Wifi
- Stove
- Microwave
- Refrigerator
- Private patio or Balcony
- Camp fire
- Personal care products

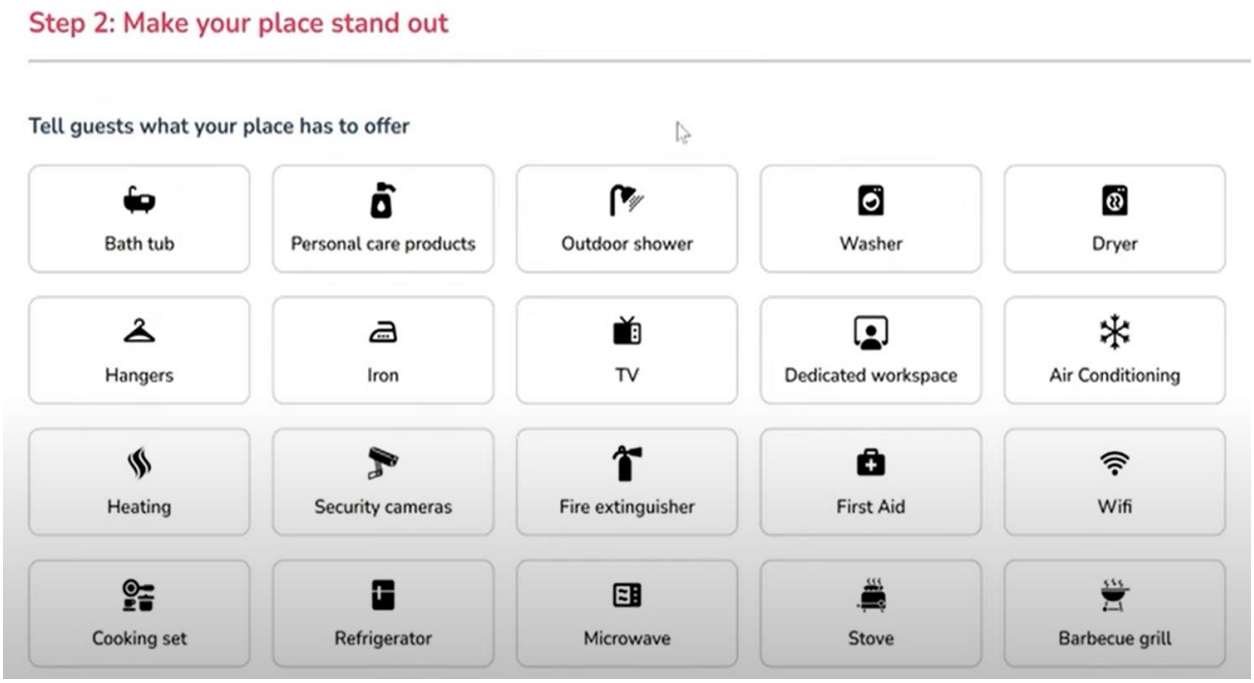
Cheerful 3 Bedroom Cottage with Fire Place

♡ Save

An entire place in Milford, Connecticut, United States

3 guests - 2 bedroom - 2 bed - 1 bath


Publish Your Place
Step 1: Tell us about your place
Which of these categories best describes your place?
All
Beachfront
Windmills
Iconic cities
Countryside
Amazing Pools
Islands
Lakefront
Ski-in/out
Castles
Caves
Camping
Arctic
Desert
Barns
Luxury


Step 2: Make your place stand out
Tell guests what your place has to offer
Bath tub
Personal care products
Outdoor shower
Washer
Dryer
Hangers
Iron
TV
Dedicated workspace
Air Conditioning
Heating
Security cameras
Fire extinguisher
First Aid
Wifi
Cooking set
Refrigerator
Microwave
Stove
Barbecue grill

Description

If it's luxury glamping you're looking for then this is the perfect fit for you. The Crooked Arrow Glampsite is quaintly nestled in the woods and offers a place to unwind & disconnect from everyday life. Spend

Highlight

Tucked away in a private wooded area behind Dutch Cousin

Highlight details

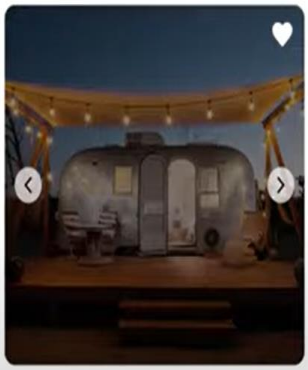
You are welcome to use all the campground has to offer including laundry, a love seat swing, and more. The camp office has ice cream, drink and other camping related items for sale.

Now, set your PRICE

$ 100

CREATE YOUR LISTING

Your Property List



Toronto, Ontario, Canada
Camping
An entire place
$500 per night

SOURCE CODE

SOFTWARE REQUIRED

- Html
- JavaScript
- Bootstrap
- React

SOFTWARE USED

❖ CLIENT

HTML

Index.html

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <meta name="theme-color" content="#000000" />

    <meta

      name="description"

      content="Web site created using create-react-app"

    />

    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
```

```
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Nunito:wght@300;400;500;600;7
00;800;900&display=swap" rel="stylesheet">
    <title>Dream Nest</title>
  </head>
  <body>
    <div id="root"></div>


  </body>
</html>
```

JSON File

Package. json

```
{
  "name": "client",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@emotion/react": "^11.11.1",
```

```
    "@emotion/styled": "^11.11.0",
    "@mui/icons-material": "^5.14.18",
    "@mui/material": "^5.14.18",
    "@reduxjs/toolkit": "^1.9.7",
    "@testing-library/jest-dom": "^5.17.0",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "react": "^18.2.0",
    "react-beautiful-dnd": "^13.1.1",
    "react-date-range": "^1.4.0",
    "react-dom": "^18.2.0",
    "react-icons": "^4.12.0",
    "react-redux": "^8.1.3",
    "react-router-dom": "^6.19.0",
    "react-scripts": "5.0.1",
    "redux-persist": "^6.0.0",
    "sass": "^1.69.5",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
```

```
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

❖ SERVER

JAVASCRIPT

index.js

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

const app = express();
```

```
const mongoose = require("mongoose");
const dotenv = require("dotenv").config();
const cors = require("cors");

const authRoutes = require("./routes/auth.js")
const listingRoutes = require("./routes/listing.js")
const bookingRoutes = require("./routes/booking.js")
const userRoutes = require("./routes/user.js")

app.use(cors());
app.use(express.json());
app.use(express.static("public"));

/* ROUTES */
app.use("/auth", authRoutes)
app.use("/properties", listingRoutes)
app.use("/bookings", bookingRoutes)
app.use("/users", userRoutes)

/* MONGOOSE SETUP */
```

```
const PORT = 3001;

mongoose
  .connect(process.env.MONGO_URL, {
    dbName: "Dream_Nest",
    useNewUrlParser: true,
    useUnifiedTopology: true,
  })
  .then(() => {
    app.listen(PORT, () => console.log(`Server Port: ${PORT}`));
  })
  .catch((err) => console.log(`${err} did not connect`));
```

auth.js

```
const router = require("express").Router();

const bcrypt = require("bcryptjs");

const jwt = require("jsonwebtoken");

const multer = require("multer");


const User = require("../models/User");
```

```
/* Configuration Multer for File Upload */
const storage = multer.diskStorage({
  destination: function (req, file, cb) {
    cb(null, "public/uploads/"); // Store uploaded files in the 'uploads' folder
  },
  filename: function (req, file, cb) {
    cb(null, file.originalname); // Use the original file name
  },
});

const upload = multer({ storage });

/* USER REGISTER */
router.post("/register", upload.single("profileImage"), async (req, res) => {
  try {
    /* Take all information from the form */
    const { firstName, lastName, email, password } = req.body;

    /* The uploaded file is available as req.file */
```

```
const profileImage = req.file;

if (!profileImage) {
  return res.status(400).send("No file uploaded");
}

/* path to the uploaded profile photo */
const profileImagePath = profileImage.path;

/* Check if user exists */
const existingUser = await User.findOne({ email });
if (existingUser) {
  return res.status(409).json({ message: "User already exists!" });
}

/* Hass the password */
const salt = await bcrypt.genSalt();
const hashedPassword = await bcrypt.hash(password, salt);

/* Create a new User */
```

```
    const newUser = new User({
      firstName,
      lastName,
      email,
      password: hashedPassword,
      profileImagePath,
    });

    /* Save the new User */
    await newUser.save();

    /* Send a successful message */
    res
      .status(200)
      .json({ message: "User registered successfully!", user: newUser });
  } catch (err) {
    console.log(err);
    res
      .status(500)
      .json({ message: "Registration failed!", error: err.message });
```

```
  }
});

/* USER LOGIN*/
router.post("/login", async (req, res) => {
  try {
    /* Take the infomation from the form */
    const { email, password } = req.body

    /* Check if user exists */
    const user = await User.findOne({ email });
    if (!user) {
      return res.status(409).json({ message: "User doesn't exist!" });
    }

    /* Compare the password with the hashed password */
    const isMatch = await bcrypt.compare(password, user.password)
    if (!isMatch) {
      return res.status(400).json({ message: "Invalid Credentials!"})
    }
```

```
    /* Generate JWT token */
    const token = jwt.sign({ id: user._id }, process.env.JWT_SECRET)
    delete user.password

    res.status(200).json({ token, user })

  } catch (err) {
    console.log(err)
    res.status(500).json({ error: err.message })
  }
})

module.exports = router
```

booking.js

```
const router = require("express").Router()

const Booking = require("../models/Booking")
```

```
/* CREATE BOOKING */
router.post("/create", async (req, res) => {
  try {
    const { customerId, hostId, listingId, startDate, endDate, totalPrice } = req.body
    const newBooking = new Booking({ customerId, hostId, listingId, startDate, endDate, totalPrice })
    await newBooking.save()
    res.status(200).json(newBooking)
  } catch (err) {
    console.log(err)
    res.status(400).json({ message: "Fail to create a new Booking!", error: err.message })
  }
})

module.exports = router
```

listing.js

```
const router = require("express").Router();
const multer = require("multer");
```

```
const Listing = require("../models/Listing");
const User = require("../models/User")

/* Configuration Multer for File Upload */
const storage = multer.diskStorage({
  destination: function (req, file, cb) {
    cb(null, "public/uploads/"); // Store uploaded files in the 'uploads' folder
  },
  filename: function (req, file, cb) {
    cb(null, file.originalname); // Use the original file name
  },
});

const upload = multer({ storage });

/* CREATE LISTING */
router.post("/create", upload.array("listingPhotos"), async (req, res) => {
  try {
    /* Take the information from the form */
```

```
const {
  creator,
  category,
  type,
  streetAddress,
  aptSuite,
  city,
  province,
  country,
  guestCount,
  bedroomCount,
  bedCount,
  bathroomCount,
  amenities,
  title,
  description,
  highlight,
  highlightDesc,
  price,
} = req.body;
```

```
const listingPhotos = req.files

if (!listingPhotos) {
  return res.status(400).send("No file uploaded.")
}

const listingPhotoPaths = listingPhotos.map((file) => file.path)

const newListing = new Listing({
  creator,
  category,
  type,
  streetAddress,
  aptSuite,
  city,
  province,
  country,
  guestCount,
  bedroomCount,
```

```
      bedCount,
      bathroomCount,
      amenities,
      listingPhotoPaths,
      title,
      description,
      highlight,
      highlightDesc,
      price,
    })

    await newListing.save()

    res.status(200).json(newListing)
  } catch (err) {
    res.status(409).json({ message: "Fail to create Listing", error: err.message })
    console.log(err)
  }
});
```

```
/* GET lISTINGS BY CATEGORY */
router.get("/", async (req, res) => {
  const qCategory = req.query.category

  try {
    let listings
    if (qCategory) {
      listings = await Listing.find({ category: qCategory }).populate("creator")
    } else {
      listings = await Listing.find().populate("creator")
    }

    res.status(200).json(listings)
  } catch (err) {
    res.status(404).json({ message: "Fail to fetch listings", error: err.message })
    console.log(err)
  }
})

/* GET LISTINGS BY SEARCH */
```

```
router.get("/search/:search", async (req, res) => {
  const { search } = req.params

  try {
    let listings = []

    if (search === "all") {
      listings = await Listing.find().populate("creator")
    } else {
      listings = await Listing.find({
        $or: [
          { category: {$regex: search, $options: "i" } },
          { title: {$regex: search, $options: "i" } },
        ]
      }).populate("creator")
    }

    res.status(200).json(listings)
  } catch (err) {
    res.status(404).json({ message: "Fail to fetch listings", error: err.message })
```

```
    console.log(err)
  }
})

/* LISTING DETAILS */
router.get("/:listingId", async (req, res) => {
  try {
    const { listingId } = req.params
    const listing = await Listing.findById(listingId).populate("creator")
    res.status(202).json(listing)
  } catch (err) {
    res.status(404).json({ message: "Listing can not found!", error: err.message })
  }
})

module.exports = router
```

user.js

```
const router = require("express").Router()
```

```
const Booking = require("../models/Booking")
const User = require("../models/User")
const Listing = require("../models/Listing")

/* GET TRIP LIST */
router.get("/:userId/trips", async (req, res) => {
  try {
    const { userId } = req.params
    const trips = await Booking.find({ customerId: userId }).populate("customerId hostId listingId")
    res.status(202).json(trips)
  } catch (err) {
    console.log(err)
    res.status(404).json({ message: "Can not find trips!", error: err.message })
  }
})

/* ADD LISTING TO WISHLIST */
router.patch("/:userId/:listingId", async (req, res) => {
  try {
```

```
    const { userId, listingId } = req.params
    const user = await User.findById(userId)
    const listing = await Listing.findById(listingId).populate("creator")

    const favoriteListing = user.wishList.find((item) => item._id.toString() === listingId)

    if (favoriteListing) {
      user.wishList = user.wishList.filter((item) => item._id.toString() !== listingId)
      await user.save()
      res.status(200).json({ message: "Listing is removed from wish list", wishList: user.wishList})
    } else {
      user.wishList.push(listing)
      await user.save()
      res.status(200).json({ message: "Listing is added to wish list", wishList: user.wishList})
    }
  } catch (err) {
    console.log(err)
```

```
    res.status(404).json({ error: err.message })
  }
})

/* GET PROPERTY LIST */
router.get("/:userId/properties", async (req, res) => {
  try {
    const { userId } = req.params
    const properties = await Listing.find({ creator: userId }).populate("creator")
    res.status(202).json(properties)
  } catch (err) {
    console.log(err)
    res.status(404).json({ message: "Can not find properties!", error: err.message })
  }
})

/* GET RESERVATION LIST */
router.get("/:userId/reservations", async (req, res) => {
  try {
    const { userId } = req.params
```

```
    const reservations = await Booking.find({ hostId: userId
}).populate("customerId hostId listingId")
    res.status(202).json(reservations)
  } catch (err) {
    console.log(err)
    res.status(404).json({ message: "Can not find reservations!", error: err.message
})
  }
})


module.exports = router
```

JSON

Package.json

```
{
  "dependencies": {
    "bcryptjs": "^2.4.3",
    "body-parser": "^1.20.2",
    "cors": "^2.8.5",
```

```
    "dotenv": "^16.3.1",
    "express": "^4.18.2",
    "jsonwebtoken": "^9.0.2",
    "mongoose": "^8.0.1",
    "multer": "^1.4.4",
    "multer-gridfs-storage": "^5.0.2",
    "nodemon": "^3.1.0"
  },
  "name": "server",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "start": "nodemon index.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

Package-lock.json:


```
{
  "name": "server",
  "version": "1.0.0",
  "lockfileVersion": 3,
  "requires": true,
  "packages": {
    "": {
      "name": "server",
      "version": "1.0.0",
      "license": "ISC",
      "dependencies": {
        "bcryptjs": "^2.4.3",
        "body-parser": "^1.20.2",
        "cors": "^2.8.5",
        "dotenv": "^16.3.1",
        "express": "^4.18.2",
        "jsonwebtoken": "^9.0.2",
        "mongoose": "^8.0.1",
        "multer": "^1.4.4",
```

```
      "multer-gridfs-storage": "^5.0.2",
      "nodemon": "^3.1.0"
    },
    "devDependencies": {}
  },
  "node_modules/@mongodb-js/saslprep": {
    "version": "1.1.1",
    "resolved": "https://registry.npmjs.org/@mongodb-js/saslprep/-/saslprep-1.1.1.tgz",
    "integrity": "sha512-t7c5K033joZZMspnHg/gWPE4kandgc2OxE74aYOtGKfgB9VPuVJPix0H6fhmm2erj5PBJ21mqcx34lpIGtUCsQ==",
    "dependencies": {
      "sparse-bitfield": "^3.0.3"
    }
  },
  "node_modules/@types/body-parser": {
    "version": "1.19.5",
    "resolved": "https://registry.npmjs.org/@types/body-parser/-/body-parser-1.19.5.tgz",
```

"integrity": "sha512-fB3Zu92ucau0iQ0JMCFQE7b/dv8Ot07NI3KaZIkIUNXq82k4eBAqUaneXfleGY9JWskeS9y+u0nXMyspcuQrCg==",
"dependencies": {
"@types/connect": "*",
"@types/node": "*"
}
},
"node_modules/@types/bson": {
"version": "4.2.0",
"resolved": "https://registry.npmjs.org/@types/bson/-/bson-4.2.0.tgz",
"integrity": "sha512-ELCPqAdroMdcuxqwMgUpifQyRoTpyYCNr1V9xKyF40VsBobsj+BbWNRvwGchMgBPGqkw655ypkjj2MEF5ywVwg==",
"deprecated": "This is a stub types definition. bson provides its own type definitions, so you do not need this installed.",
"dependencies": {
"bson": "*"
}
},
"node_modules/@types/connect": {

```
    "version": "3.4.38",
    "resolved": "https://registry.npmjs.org/@types/connect/-/connect-3.4.38.tgz",
    "integrity": "sha512-
K6uROf1LD88uDQqJCktA4yzL1YYAK6NgfsI0v/mTgyPKWsX1CnJ0XPSDhVie
jru1GcRkLWb8RlzFYJRqGUbaug==",
    "dependencies": {
      "@types/node": "*"
    }
  },
  "node_modules/@types/express": {
    "version": "4.17.21",
    "resolved": "https://registry.npmjs.org/@types/express/-/express-4.17.21.tgz",
    "integrity": "sha512-
ejlPM315qwLpaQlQDTjPdsUFSc6ZsP4AN6AlWnogPjQ7CVi7PYF3YVz+CY3j
E2pwYf7E/7HlDAN0rV2GxTG0HQ==",
    "dependencies": {
      "@types/body-parser": "*",
      "@types/express-serve-static-core": "^4.17.33",
      "@types/qs": "*",
      "@types/serve-static": "*"
    }
```

```
    },
    "node_modules/@types/express-serve-static-core": {
      "version": "4.17.41",
      "resolved": "https://registry.npmjs.org/@types/express-serve-static-core/-
/express-serve-static-core-4.17.41.tgz",
      "integrity": "sha512-
OaJ7XLaelTgrvlZD8/aa0vvvxZdUmlCn6MtWeB7TkiKW70BQLc9XEPpDLPdbo
52ZhXUCrznlWdCHWxJWtdyajA==",
      "dependencies": {
        "@types/node": "*",
        "@types/qs": "*",
        "@types/range-parser": "*",
        "@types/send": "*"
      }
    },
    "node_modules/@types/http-errors": {
      "version": "2.0.4",
      "resolved": "https://registry.npmjs.org/@types/http-errors/-/http-errors-
2.0.4.tgz",
```

"integrity": "sha512-
D0CFMMtydbJAegzOyHjtiKPLlvnm3iTZyZRSZoLq2mRhDdmLfIWOCYPfQJ4
cu2erKghU++QvjcUjp/5h7hESpA=="

},

"node_modules/@types/mime": {

"version": "1.3.5",

"resolved": "https://registry.npmjs.org/@types/mime/-/mime-1.3.5.tgz",

"integrity": "sha512-
/pyBZWSLD2n0dcHE3hq8s8ZvcETHtEuF+3E7XVt0Ig2nvsVQXdghHVcEkIWjy
9A0wKfTn97a/PSDYohKIlnP/w=="

},

"node_modules/@types/mongodb": {

"version": "3.6.20",

"resolved": "https://registry.npmjs.org/@types/mongodb/-/mongodb-
3.6.20.tgz",

"integrity": "sha512-
WcdpPJCakFzcWWD9juKoZbRtQxKIMYF/JIAM4JrNHrMcnJL6/a2NWjXxW7f
o9hxboxxkg+icff8d7+WIEvKgYQ==",

"dependencies": {

"@types/bson": "*",

"@types/node": "*"

```
    }
  },
  "node_modules/@types/multer": {
    "version": "1.4.10",
    "resolved": "https://registry.npmjs.org/@types/multer/-/multer-1.4.10.tgz",
    "integrity": "sha512-
6l9mYMhUe8wbnz/67YIjc7ZJyQNZoKq7fRXVf7nMdgWgalD0KyzJ2ywI7hoAT
USXSbTu9q2HBiEwzy0tNN1v2w==",
    "dependencies": {
      "@types/express": "*"
    }
  },
  "node_modules/@types/node": {
    "version": "20.9.1",
    "resolved": "https://registry.npmjs.org/@types/node/-/node-20.9.1.tgz",
    "integrity": "sha512-
HhmzZh5LSJNS5O8jQKpJ/3ZcrrlG6L70hpGqMIAoM9YVD0YBRNWYsfwcXq
8VnSjlNpCpgLzMXdiPo+dxcvSmiA==",
    "dependencies": {
      "undici-types": "~5.26.4"
    }
```

```
  },
  "node_modules/@types/pump": {
    "version": "1.1.3",
    "resolved": "https://registry.npmjs.org/@types/pump/-/pump-1.1.3.tgz",
    "integrity": "sha512-
ZyooTTivmOwPfOwLVaszkF8Zq6mvavgjuHYitZhrIjfQAJDH+kIP3N+MzpG1zD
AslsHvVz6Q8ECfivix3qLJaQ==",
    "dependencies": {
      "@types/node": "*"
    }
  },
  "node_modules/@types/qs": {
    "version": "6.9.10",
    "resolved": "https://registry.npmjs.org/@types/qs/-/qs-6.9.10.tgz",
    "integrity": "sha512-
3Gnx08Ns1sEoCrWssEgTSJs/rsT2vhGP+Ja9cnnk9k4ALxinORlQneLXFeFKOTJ
MOeZUFD1s7w+w2AphTpvzZw=="
  },
  "node_modules/@types/range-parser": {
    "version": "1.2.7",
```

"resolved": "https://registry.npmjs.org/@types/range-parser/-/range-parser-1.2.7.tgz",

"integrity": "sha512-hKormJbkJqzQGhziax5PItDUTMAM9uE2XXQmM37dyd4hVM+5aVl7oVxMVUiVQn2oCQFN/LKCZdvSM0pFRqbSmQ=="

},

"node_modules/@types/send": {

"version": "0.17.4",

"resolved": "https://registry.npmjs.org/@types/send/-/send-0.17.4.tgz",

"integrity": "sha512-x2EM6TJOybec7c52BX0ZspPodMsQUd5L6PRwOunVyVUhXiBSKf3AezDL8Dgvgt5o0UfKNfuA0eMLr2wLT4AiBA==",

"dependencies": {

"@types/mime": "^1",

"@types/node": "*"

}

},

"node_modules/@types/serve-static": {

"version": "1.15.5",

"resolved": "https://registry.npmjs.org/@types/serve-static/-/serve-static-1.15.5.tgz",

"integrity": "sha512-PDRk21MnK70hja/YF8AHfC7yIsiQHn1rcXx7ijCFBX/k+XQJhQT/gw3xekXKJvx+5SXaMMS8oqQy09Mzvz2TuQ==",

"dependencies": {

"@types/http-errors": "*",

"@types/mime": "*",

"@types/node": "*"

}

},

"node_modules/@types/webidl-conversions": {

"version": "7.0.3",

"resolved": "https://registry.npmjs.org/@types/webidl-conversions/-/webidl-conversions-7.0.3.tgz",

"integrity": "sha512-CiJJvcRtIgzadHCYXw7dqEnMNRjhGZlYK05Mj9OyktqV8uVT8fD2BFOB7S1uwBE3Kj2Z+4UyPmFw/Ixgw/LAlA=="

},

"node_modules/@types/whatwg-url": {

"version": "8.2.2",

"resolved": "https://registry.npmjs.org/@types/whatwg-url/-/whatwg-url-8.2.2.tgz",

```
    "integrity": "sha512-
FtQu10RWgn3D9U4aazdwIE2yzphmTJREDqNdODHrbrZmmMqI0vMheC/6NE/
J1Yveaj8H+ela+YwWTjq5PGmuhA==",
    "dependencies": {
      "@types/node": "*",
      "@types/webidl-conversions": "*"
    }
  },
  "node_modules/abbrev": {
    "version": "1.1.1",
    "resolved": "https://registry.npmjs.org/abbrev/-/abbrev-1.1.1.tgz",
    "integrity": "sha512-
nne9/IiQ/hzIhY6pdDnbBtz7DjPTKrY00P/zvPSm5pOFkl6xuGrGnXn/VtTNNfNt
AfZ9/1RtehkszU9qcTii0Q=="
  },
  "node_modules/accepts": {
    "version": "1.3.8",
    "resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.8.tgz",
    "integrity": "sha512-
PYAthTa2m2VKxuvSD3DPC/Gy+U+sOA1LAuT8mkmRuvw+NACSaeXEQ+NH
cVF7rONl6qcaxV3Uuemwawk+7+SJLw==",
```

```
    "dependencies": {
      "mime-types": "~2.1.34",
      "negotiator": "0.6.3"
    },
    "engines": {
      "node": ">= 0.6"
    }
  },
  "node_modules/anymatch": {
    "version": "3.1.3",
    "resolved": "https://registry.npmjs.org/anymatch/-/anymatch-3.1.3.tgz",
    "integrity": "sha512-
KMReFUr0B4t+D+OBkjR3KYqvocp2XaSzO55UcB6mgQMd3KbcE+mWTyvVV
7D/zsdEbNnV6acZUutkiHQXvTr1Rw==",
    "dependencies": {
      "normalize-path": "^3.0.0",
      "picomatch": "^2.0.4"
    },
    "engines": {
      "node": ">= 8"
```

```
    }
  },
  "node_modules/append-field": {
    "version": "1.0.0",
    "resolved": "https://registry.npmjs.org/append-field/-/append-field-1.0.0.tgz",
    "integrity": "sha512-
klpgFSWLW1ZEs8svjfb7g4qWY0YS5imI82dTg+QahUvJ8YqAY0P10Uk8tTyh9Z
GuYEZEMaeJYCF5BFuX552hsw=="
  },
  "node_modules/array-flatten": {
    "version": "1.1.1",
    "resolved": "https://registry.npmjs.org/array-flatten/-/array-flatten-1.1.1.tgz",
    "integrity": "sha512-
PCVAQswWemu6UdxsDFFX/+gVeYqKAod3D3UVm91jHwynguOwAvYPhx8nN
lM++NqRcK6CxxpUafjmhIdKiHibqg=="
  },
  "node_modules/balanced-match": {
    "version": "1.0.2",
    "resolved": "https://registry.npmjs.org/balanced-match/-/balanced-match-
1.0.2.tgz",
```

```
    "integrity": "sha512-
3oSeUO0TMV67hN1AmbXsK4yaqU7tjiHlbxRDZOpH0KW9+CeX4bRAaX0An
xt0tx2MrpRpWwQaPwIlISEJhYU5Pw=="
  },
  "node_modules/bcryptjs": {
    "version": "2.4.3",
    "resolved": "https://registry.npmjs.org/bcryptjs/-/bcryptjs-2.4.3.tgz",
    "integrity": "sha512-
V/Hy/X9Vt7f3BbPJEi8BdVFMByHi+jNXrYkW3huaybV/kQ0KJg0Y6PkEMbn+z
eT+i+SiKZ/HMqJGIIt4LZDqNQ=="
  },
  "node_modules/binary-extensions": {
    "version": "2.3.0",
    "resolved": "https://registry.npmjs.org/binary-extensions/-/binary-extensions-
2.3.0.tgz",
    "integrity": "sha512-
Ceh+7ox5qe7LJuLHoY0feh3pHuUDHAcRUeyL2VYghZwfpkNIy/+8Ocg0a3UuS
oYzavmylwuLWQOf3hl0jjMMIw==",
    "engines": {
      "node": ">=8"
    },
```

```
    "funding": {
      "url": "https://github.com/sponsors/sindresorhus"
    }

    "funding": {
      "url": "https://github.com/sponsors/ljharb"
    }
  },
  "node_modules/sift": {
    "version": "16.0.1",
    "resolved": "https://registry.npmjs.org/sift/-/sift-16.0.1.tgz",
    "integrity": "sha512-
Wv6BjQ5zbhW7VFefWusVP33T/EM0vYikCaQ2qR8yULbsilAT8/wQaXvuQ3pt
GLpoKx+lihJE3y2UTgKDyyNHZQ=="
  },
  "node_modules/simple-update-notifier": {
    "version": "2.0.0",
    "resolved": "https://registry.npmjs.org/simple-update-notifier/-/simple-update-
notifier-2.0.0.tgz",
```

```
    "integrity": "sha512-
a2B9Y0KlNXl9u/vsW6sTIu9vGEpfKu2wRV6l1H3XEas/0gUIzGzBoP/IouTcUQb
m9JWZLH3COxyn03TYlFax6w==",
    "dependencies": {
      "semver": "^7.5.3"
    },
    "engines": {
      "node": ">=10"
    }
  },
  "node_modules/sparse-bitfield": {
    "version": "3.0.3",
    "resolved": "https://registry.npmjs.org/sparse-bitfield/-/sparse-bitfield-
3.0.3.tgz",
    "integrity": "sha512-
kvzhi7vqKTfkh0PZU+2D2PIllw2ymqJKujUcyPMd9Y75Nv4nPbGJZXNhxsgdQa
b2BmlDct1YnfQCguEvHr7VsQ==",
    "dependencies": {
      "memory-pager": "^1.0.2"
    }
  },
```

```
    "node_modules/statuses": {
      "version": "2.0.1",
      "resolved": "https://registry.npmjs.org/statuses/-/statuses-2.0.1.tgz",
      "integrity": "sha512-
RwNA9Z/7PrK06rYLIzFMlaF+l73iwpzsqRIFgbMLbTcLD6cOao82TaWefPXQv
B2fOC4AjuYSEndS7N/mTCbkdQ==",
      "engines": {
        "node": ">= 0.8"
      }
    },
    "node_modules/streamsearch": {
      "version": "0.1.2",
      "resolved": "https://registry.npmjs.org/streamsearch/-/streamsearch-0.1.2.tgz",
      "integrity": "sha512-
jos8u++JKm0ARcSUTAZXOVC0mSox7Bhn6sBgty73P1f3JGf7yG2clTbBNHUd
de/kdvP2FESam+vM6l8jBrNxHA==",
      "engines": {
        "node": ">=0.8.0"
      }
    },
    "node_modules/string_decoder": {
```

```
    "version": "0.10.31",
    "resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-0.10.31.tgz",
    "integrity": "sha512-ev2QzSzWPYmy9GuqfIVildA4OdcGLeFZQrq5ys6RtiuF+RQQiZWr8TZNyAcuVXyQRYfEO+MsoB/1BuQVhOJuoQ=="
  },
  "node_modules/supports-color": {
    "version": "5.5.0",
    "resolved": "https://registry.npmjs.org/supports-color/-/supports-color-5.5.0.tgz",
    "integrity": "sha512-QjVjwdXIt408MIiAqCX4oUKsgU2EqAGzs2Ppkm4aQYbjm+ZEWEcW4SfFNTr4uMNZma0ey4f5lgLrkB0aX0QMow==",
    "dependencies": {
      "has-flag": "^3.0.0"
    },
    "engines": {
      "node": ">=4"
    }
  },
```

```
    "node_modules/to-regex-range": {
      "version": "5.0.1",
      "resolved": "https://registry.npmjs.org/to-regex-range/-/to-regex-range-5.0.1.tgz",
      "integrity": "sha512-65P7iz6X5yEr1cwcgvQxbbIw7Uk3gOy5dIdtZ4rDveLqhrdJP+Li/Hx6tyK0NEb+2GCyneCMJiGqrADCSNk8sQ==",
      "dependencies": {
        "is-number": "^7.0.0"
      },
      "engines": {
        "node": ">=8.0"
      }
    },
    "node_modules/toidentifier": {
      "version": "1.0.1",
      "resolved": "https://registry.npmjs.org/toidentifier/-/toidentifier-1.0.1.tgz",
      "integrity": "sha512-o5sSPKEkg/DIQNmH43V0/uerLrpzVedkUh8tGNvaeXpfpuwjKenlSox/2O/BTlZUtEe+JG7s5YhEz608PlAHRA==",
      "engines": {
```

```
      "node": ">=0.6"
    }
  },
  "node_modules/touch": {
    "version": "3.1.0",
    "resolved": "https://registry.npmjs.org/touch/-/touch-3.1.0.tgz",
    "integrity": "sha512-
WBx8Uy5TLtOSRtIq+M03/sKDrXCLHxwDcquSP2c43Le03/9serjQBIztjRz6FkJ
ez9D/hleyAXTBGLwwZUw9lA==",
    "dependencies": {
      "nopt": "~1.0.10"
    },
    "bin": {
      "nodetouch": "bin/nodetouch.js"
    }
  },
  "node_modules/tr46": {
    "version": "3.0.0",
    "resolved": "https://registry.npmjs.org/tr46/-/tr46-3.0.0.tgz",
```

"integrity": "sha512-
l7FvfAHlcmulp8kr+flpQZmVwtu7nfRV7NZujtN0OqES8EL4O4e0qqzL0DC5gAv
x/ZC/9lk6rhcUwYvkBnBnYA==",
"dependencies": {
"punycode": "^2.1.1"
},
"engines": {
"node": ">=12"
}
},
"node_modules/type-is": {
"version": "1.6.18",
"resolved": "https://registry.npmjs.org/type-is/-/type-is-1.6.18.tgz",
"integrity": "sha512-
TkRKr9sUTxEH8MdfuCSP7VizJyzRNMjj2J2do2Jr3Kym598JVdEksuzPQCnlFP
W4ky9Q+iA+ma9BGm06XQBy8g==",
"dependencies": {
"media-typer": "0.3.0",
"mime-types": "~2.1.24"
},
"engines": {

```
      "node": ">= 0.6"
    }
  },
  "node_modules/typedarray": {
    "version": "0.0.6",
    "resolved": "https://registry.npmjs.org/typedarray/-/typedarray-0.0.6.tgz",
    "integrity": "sha512-
/aCDEGatGvZ2BIk+HmLf4ifCJFwvKFNb9/JeZPMulfgFracn9QFcAf5GO8B/mw
eUjSoblS5In0cWhqpfs/5PQA=="
  },
  "node_modules/undefsafe": {
    "version": "2.0.5",
    "resolved": "https://registry.npmjs.org/undefsafe/-/undefsafe-2.0.5.tgz",
    "integrity": "sha512-
WxONCrssBM8TSPRqN5EmsjVrsv4A8X12J4ArBiiayv3DyyG3ZlIg6yysuuSYdZ
sVz3TKcTg2fd//Ujd4CHV1iA=="
  },
  "node_modules/undici-types": {
    "version": "5.26.5",
    "resolved": "https://registry.npmjs.org/undici-types/-/undici-types-5.26.5.tgz",
```

```
      "integrity": "sha512-
JlCMO+ehdEIKqlFxk6IfVoAUVmgz7cU7zD/h9XZ0qzeosSHmUJVOzSQvvYSY
WXkFXC+IfLKSIffhv0sVZup6pA=="
    },
    "node_modules/unpipe": {
      "version": "1.0.0",
      "resolved": "https://registry.npmjs.org/unpipe/-/unpipe-1.0.0.tgz",
      "integrity": "sha512-
pjy2bYhSsufwWlKwPc+l3cN7+wuJlK6uz0YdJEOlQDbl6jo/YlPi4mb8agUkVC8
BF7V8NuzeyPNqRksA3hztKQ==",
      "engines": {
        "node": ">= 0.8"
      }
    },
    "node_modules/util-deprecate": {
      "version": "1.0.2",
      "resolved": "https://registry.npmjs.org/util-deprecate/-/util-deprecate-1.0.2.tgz",
      "integrity": "sha512-
EPD5q1uXyFxJpCrLnCc1nHnq3gOa6DZBocAIiI2TaSCA7VCJ1UJDMagCzIkX
NsUYfD1daK//LTEQ8xiIbrHtcw=="
    },
```

```
    "node_modules/utils-merge": {
      "version": "1.0.1",
      "resolved": "https://registry.npmjs.org/utils-merge/-/utils-merge-1.0.1.tgz",
      "integrity": "sha512-
pMZTvIkT1d+TFGvDOqodOclx0QWkkgi6Tdoa8gC8ffGAAqz9pzPTZWAybbsH
HoED/ztMtkv/VoYTYyShUn81hA==",
      "engines": {
        "node": ">= 0.4.0"
      }
    },
    "node_modules/vary": {
      "version": "1.1.2",
      "resolved": "https://registry.npmjs.org/vary/-/vary-1.1.2.tgz",
      "integrity": "sha512-
BNGbWLfd0eUPabhkXUVm0j8uuvREyTh5ovRa/dyow/BqAbZJyC+5fU+IzQOz
mAKzYqYRAISoRhdQr3eIZ/PXqg==",
      "engines": {
        "node": ">= 0.8"
      }
    },
    "node_modules/webidl-conversions": {
```

```
    "version": "7.0.0",
    "resolved": "https://registry.npmjs.org/webidl-conversions/-/webidl-conversions-7.0.0.tgz",
    "integrity": "sha512-VwddBukDzu71offAQR975unBIGqfKZpM+8ZX6ySk8nYhVoo5CYaZyzt3YBvYtRtO+aoGlqxPg/B87NGVZ/fu6g==",
    "engines": {
      "node": ">=12"
    }
  },
  "node_modules/whatwg-url": {
    "version": "11.0.0",
    "resolved": "https://registry.npmjs.org/whatwg-url/-/whatwg-url-11.0.0.tgz",
    "integrity": "sha512-RKT8HExMpoYx4igMiVMY83lN6UeITKJlBQ+vR/8ZJ8OCdSiN3RwCq+9gH0+Xzj0+5IrM6i4j/6LuvzbZIQgEcQ==",
    "dependencies": {
      "tr46": "^3.0.0",
      "webidl-conversions": "^7.0.0"
    },
    "engines": {
```

```
      "node": ">=12"
    }
  },
  "node_modules/wrappy": {
    "version": "1.0.2",
    "resolved": "https://registry.npmjs.org/wrappy/-/wrappy-1.0.2.tgz",
    "integrity": "sha512-
l4Sp/DRseor9wL6EvV2+TuQn63dMkPjZ/sp9XkghTEbV9KlPS1xUsZ3u7/IQO4
wxtcFB4bgpQPRcR3QCvezPcQ=="
  },
  "node_modules/xtend": {
    "version": "4.0.2",
    "resolved": "https://registry.npmjs.org/xtend/-/xtend-4.0.2.tgz",
    "integrity": "sha512-
LKYU1iAXJXUgAXn9URjiu+MWhyUXHsvfp7mcuYm9dSUKK0/CjtrUwFAxD
82/mCWbtLsGjFIad0wIsod4zrTAEQ==",
    "engines": {
      "node": ">=0.4"
    }
  },
  "node_modules/yallist": {
```

```
      "version": "4.0.0",
      "resolved": "https://registry.npmjs.org/yallist/-/yallist-4.0.0.tgz",
      "integrity": "sha512-
3wdGidZyq5PB084XLES5TpOSRA3wjXAlIWMhum2kRcv/41Sn2emQ0dycQW
4uZXLejwKvg6EsvbdlVL+FYEct7A=="
    }
  }
}
```

CONCLUSION

Conclusion:

The home rental application has been successfully developed and completed, demonstrating high scalability and efficiency. It effectively addresses the challenges previously faced in manual administration within the company. Through rigorous testing with sample inputs, the system has been proven to work effectively.

Implemented within the specified environment, the system operates efficiently, meeting the needs of both tenants and landlords. Its flexibility allows for easy adaptation to changes as needed, ensuring it remains relevant and functional over time.

With a user-friendly interface and simple design, the software offers a seamless experience for users, facilitating easy navigation and interaction. Overall, the home rental application represents a significant improvement in streamlining rental processes and enhancing user satisfaction.

BIBLIOGRAPHY

Book References:

- Freeman, Julie C., et al. HTML, CSS, and JavaScript All in One: Covering HTML5, CSS3, and ES6, Sams Teach Yourself.
- Wieruch, Robin. The Road to Learn React: Your Journey to Master Plain Yet Pragmatic React. Js.
- Casciaro, Mario. Node.js Design Patterns: Design and Implement Production-grade Node.js Applications Using Proven Patterns and Techniques.

Websites:

1. **HTML:** https://www.w3schools.com/html/
2. **CSS:** https://www.w3schools.com/css/
3. **JavaScript:** https://www.geeksforgeeks.org/javascript/
4. **ReactJS:** https://legacy.reactjs.org/tutorial/tutorial.html
5. **Node.js**: https://nodejs.org/en/learn/getting-started/introduction-to-nodejs
6. **Express.js:** https://www.tutorialspoint.com/expressjs/index.htm
7. **MongoDB:** https://www.w3schools.com/mongodb/

These resources have been instrumental in the development of our home rental application, providing valuable insights, tutorials, and documentation on HTML, CSS, JavaScript, React.js, Node.js, Express.js, and MongoDB.