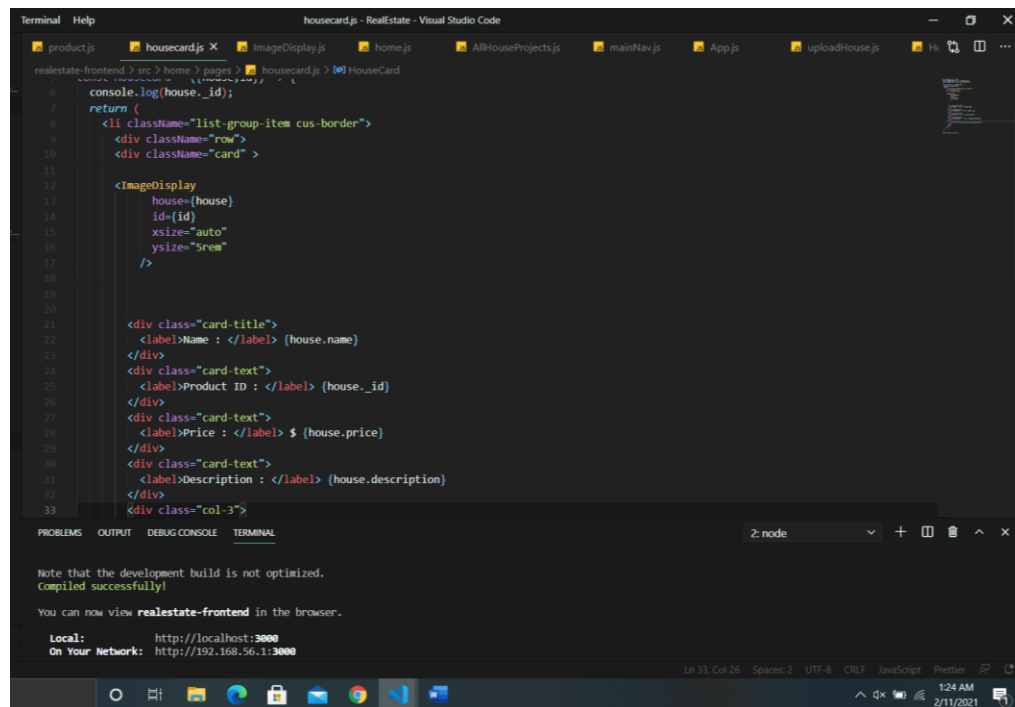


1. Describe the scenario using your own words

According to my IT number, the scenario is RealEstate. So I choose to implement a real estate house advertising website. Here the user can view the houses build by the real estate company and connect with the staff to get further details. The user must have a valid email and password to create an account. Once sign up to the website, the user can sign in using the password and email. Only the admin can add houses to the website. Since this is a small and individual app, I didn't implement many functionalities. The main functionalities are Log In, Sign In, House Project view.

2. Implementation of the main three functionalities (both frontend and backend).

House View frontend



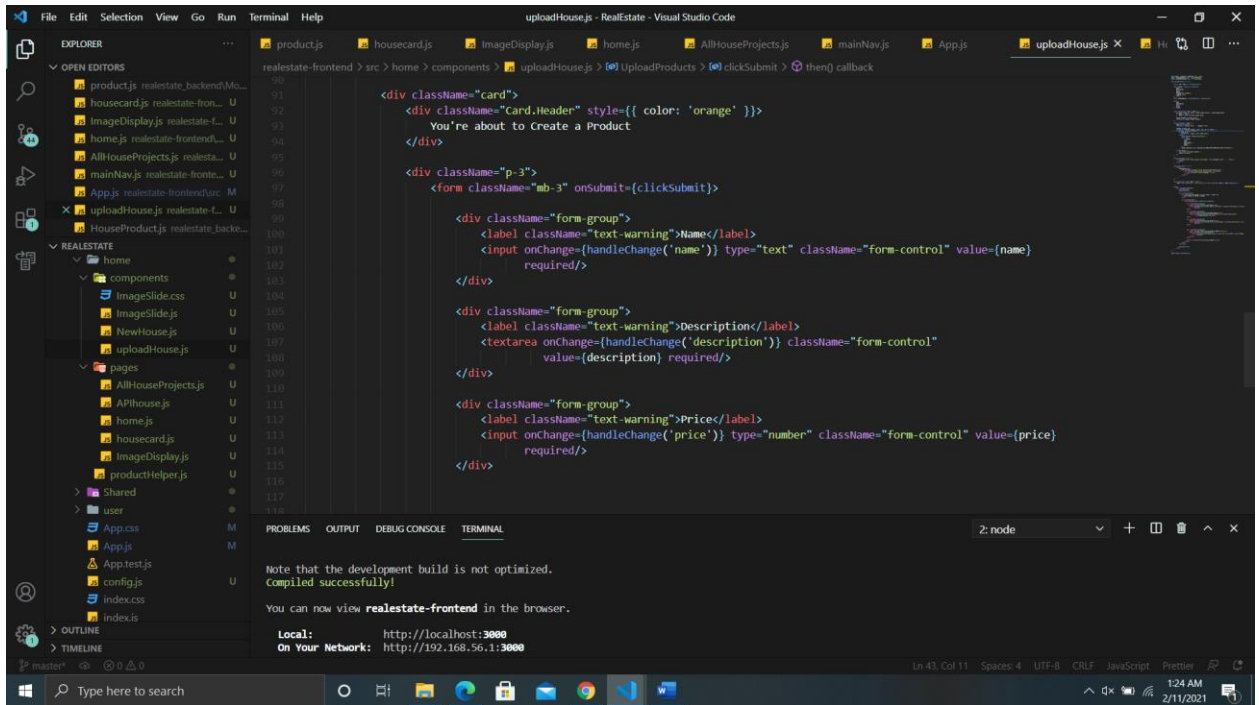
The screenshot shows the Visual Studio Code editor with the 'housecard.js' file open. The file contains a React component for displaying a house card. The code includes a console log for the house ID, a return statement with a JSX element, and an `ImageDisplay` component. The JSX element has a class name 'list-group-item cus-border' and contains a 'row' class with a 'card' class. Inside the card, there is an `ImageDisplay` component with props for house, id, xsize, and ysize. Below the image, there are four text elements: 'card-title' with a label for house.name, 'card-text' with a label for house._id, 'card-text' with a label for house.price, and 'card-text' with a label for house.description. The card is wrapped in a 'col-3' class.

```
1 console.log(house._id);
2
3 return (
4   <li className="list-group-item cus-border">
5     <div className="row">
6       <div className="card">
7
8         <ImageDisplay
9           house={house}
10          id={id}
11          xsize="auto"
12          ysize="5rem"
13        />
14
15        <div className="card-title">
16          <label>Name : </label> {house.name}
17        </div>
18        <div className="card-text">
19          <label>Product ID : </label> {house._id}
20        </div>
21        <div className="card-text">
22          <label>Price : </label> $ {house.price}
23        </div>
24        <div className="card-text">
25          <label>Description : </label> {house.description}
26        </div>
27      </div>
28    </div>
29  </li>
30 </div>
31 </div>
32 </div>
33 </div>
```

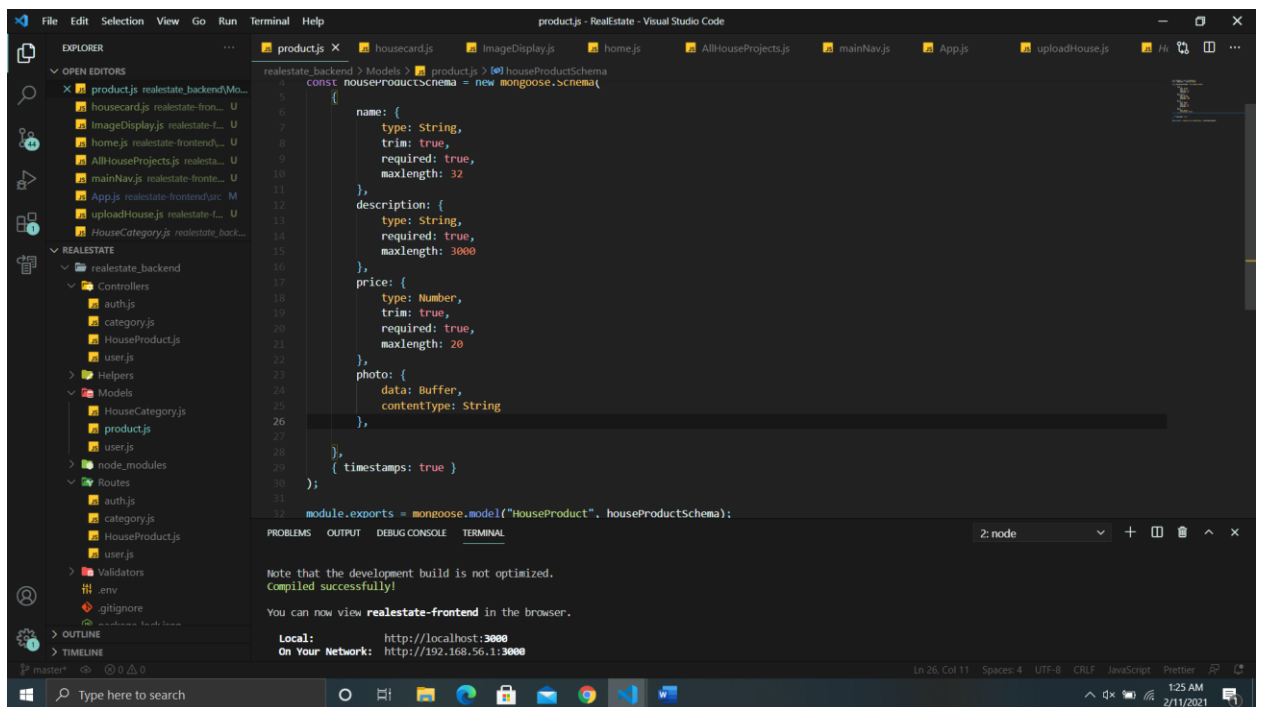
The terminal output shows the following messages:

```
Note that the development build is not optimized.
Compiled successfully!
You can now view realestate-frontend in the browser.
Local: http://localhost:3000
On Your Network: http://192.168.56.1:3000
```

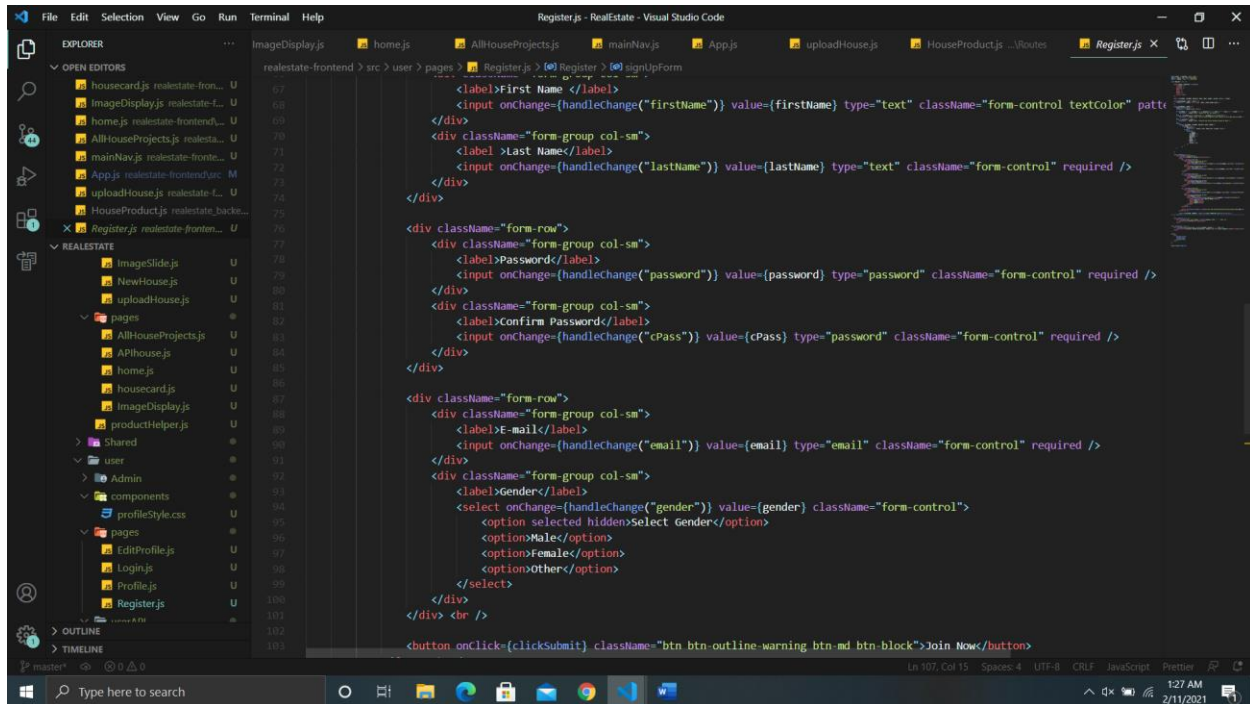
Add house(By admin) frontend



house model backend



Sign In frontend



```
File Edit Selection View Go Run Terminal Help
Register.js - RealEstate - Visual Studio Code

EXPLORER
  OPEN EDITORS
    housecard.js realstate-frontend-front... U 67
    ImageDisplay.js realstate-frontend-front... U 68
    home.js realstate-frontend-front... U 69
    AllHouseProjects.js realstate-frontend-front... U 70
    mainNav.js realstate-frontend-front... U 71
    App.js realstate-frontend-front... U 72
    uploadHouse.js realstate-frontend-front... U 73
    HouseProduct.js realstate-frontend-front... U 74
    Register.js realstate-frontend-front... U 75
  REALSTATE
    ImageSlide.js U 76
    NewHouse.js U 77
    uploadHouse.js U 78
    pages
      AllHouseProjects.js U 79
      APIHouse.js U 80
      home.js U 81
      housecard.js U 82
      ImageDisplay.js U 83
      productHelper.js U 84
    Shared
      user U 85
      Admin U 86
      components
        profileStyle.css U 87
      pages
        EditProfile.js U 88
        Login.js U 89
        Profile.js U 90
        Register.js U 91
    OUTLINE
    TIMELINE

realstate-frontend > src > user > pages > Register.js > Register.js > signUpForm
...
<label>First Name</label>
<input onChange={handleChange("firstName")} value={firstName} type="text" className="form-control" pattern="^[a-zA-Z ]+$" />
</div>
<div className="form-group col-sm">
  <label>Last Name</label>
  <input onChange={handleChange("lastName")} value={lastName} type="text" className="form-control" required />
</div>
</div>
<div className="form-row">
  <div className="form-group col-sm">
    <label>Password</label>
    <input onChange={handleChange("password")} value={password} type="password" className="form-control" required />
  </div>
  <div className="form-group col-sm">
    <label>Confirm Password</label>
    <input onChange={handleChange("cPass")} value={cPass} type="password" className="form-control" required />
  </div>
</div>
<div className="form-row">
  <div className="form-group col-sm">
    <label>E-mail</label>
    <input onChange={handleChange("email")} value={email} type="email" className="form-control" required />
  </div>
  <div className="form-group col-sm">
    <label>Gender</label>
    <select onChange={handleChange("gender")} value={gender} className="form-control">
      <option selected hidden>Select Gender</option>
      <option>Male</option>
      <option>Female</option>
      <option>Other</option>
    </select>
  </div>
</div>
<br />
<button onClick={clickSubmit} className="btn btn-outline-warning btn-md btn-block">Join Now</button>
...
Ln 107, Col 15 Spaces: 4 UTF-8 CRLF JavaScript Prettier
```

House project add backend

```
File Edit Selection View Go Run Terminal Help
HouseProduct.js - RealEstate - Visual Studio Code

EXPLORER
OPEN EDITORS
category.js realestate_backend/Rou...
housecard.js realestate-front... U
ImageDisplay.js realestate-t... U
home.js realestate-frontend... U
AllHouseProjects.js realesta... U
mainNav.js realestate-fronte... U
App.js realestate-frontend... M
uploadHouse.js realestate-t... U
HouseProduct.js realestate_backe...
REALSTATE
realestate_backend
Controllers
auth.js
category.js
HouseProduct.js
user.js
Helpers
Models
HouseCategory.js
product.js
user.js
node_modules
Routes
auth.js
category.js
HouseProduct.js
user.js
Validators
.env
.gitignore
OUTLINE
TIMELINE
master 0 0 0

realestate_backend > Routes > HouseProduct.js > ...
1 const express = require("express");
2 const router = express.Router();
3 const { requiresSignIn, isAdmin, isAuthenticated } = require("../Controllers/auth");
4 const { userId } = require("../Controllers/user");
5 const { create, productById, read, remove, update, newArrivalList, categoryRelatedProducts, listWithFilter, photo } = require("../Controller
6
7
8
9 router.post("/houseProduct/create/:userId",requiresSignIn,isAuth, isAdmin,create);
10 router.get("/houseProduct/:productId", read);
11 router.post("/houseProducts/withFilter", listWithFilter);
12 router.get("/houseProduct/photo/:productId", photo);
13
14 router.param("userId", userId);
15 router.param("productId", productById);
16
17 module.exports = router;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: node

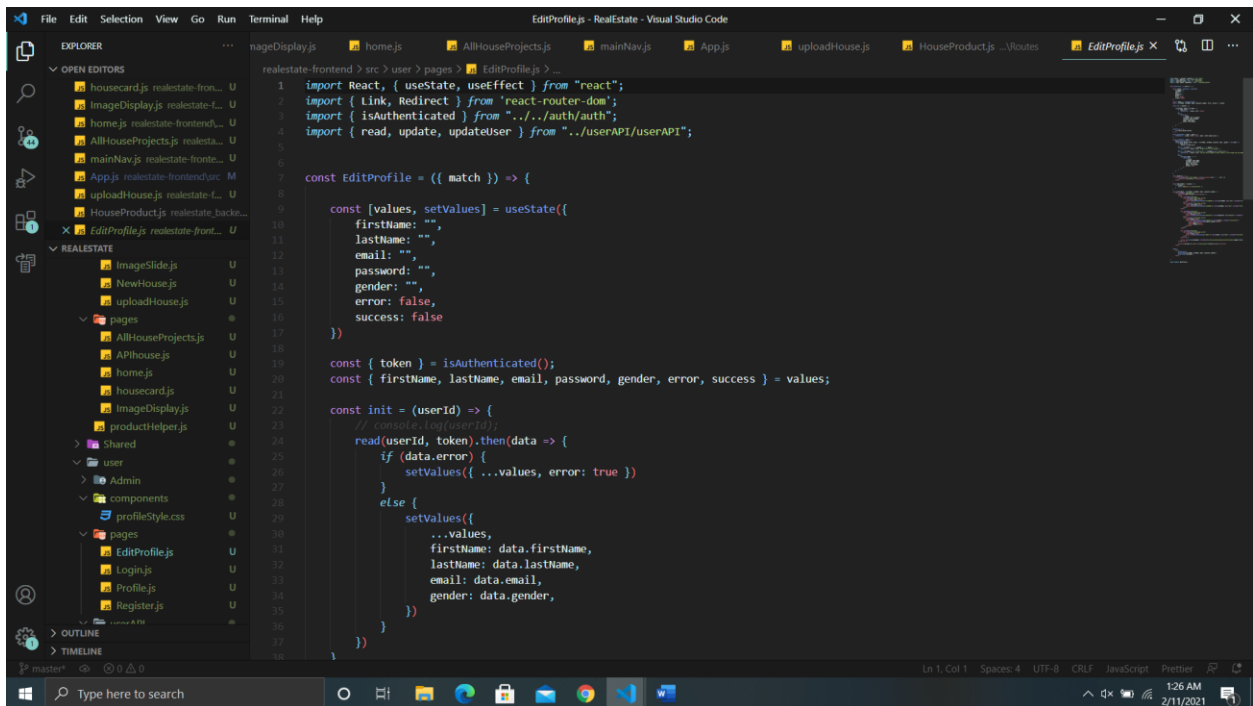
Note that the development build is not optimized.
Compiled successfully!
You can now view realestate-frontend in the browser.

Local: http://localhost:3000
On Your Network: http://192.168.56.1:3000

Ln 12, Col 26 (12 selected) Spaces: 4 UTF-8 CRLF JavaScript Prettier

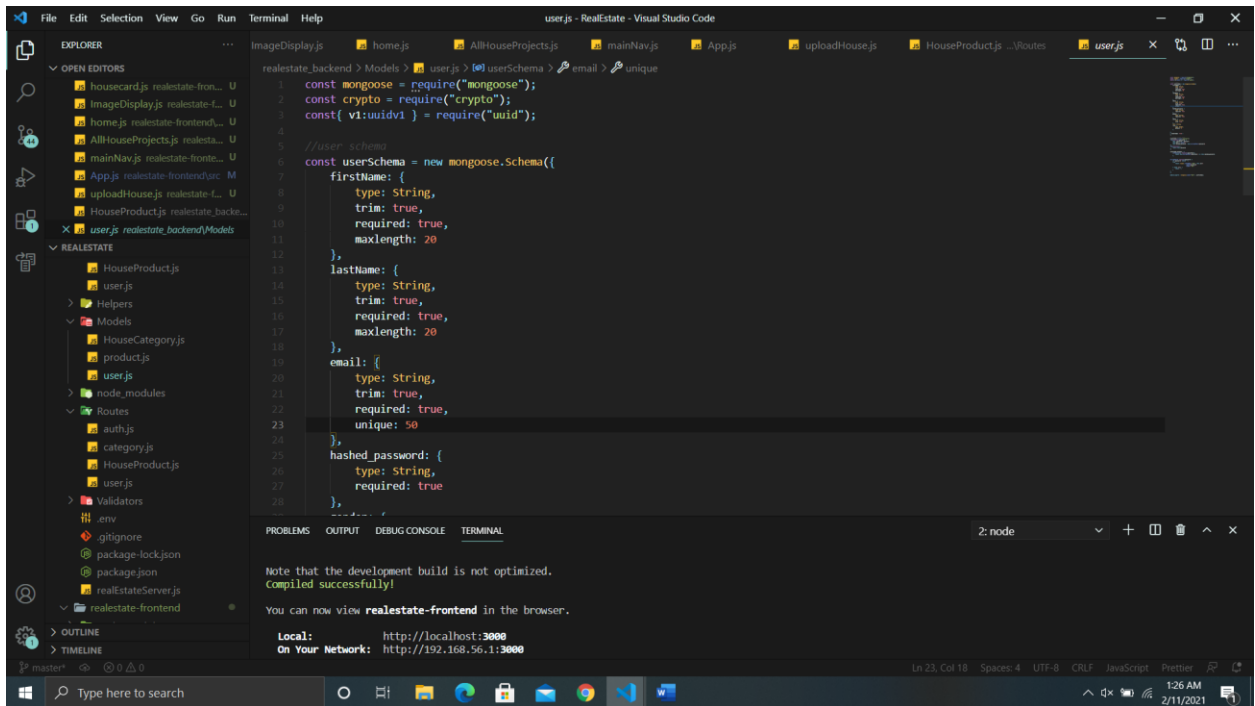
1:25 AM 2/11/2021

User profile edit frontend



```
1 import React, { useState, useEffect } from "react";
2 import { Link, Redirect } from "react-router-dom";
3 import { isAuthenticated } from "../../auth/auth";
4 import { read, update, updateUser } from "../../userAPI/userAPI";
5
6 const EditProfile = ({ match }) => {
7
8   const [values, setValues] = useState({
9     firstName: "",
10    lastName: "",
11    email: "",
12    password: "",
13    gender: "",
14    error: false,
15    success: false
16  });
17
18   const { token } = isAuthenticated();
19   const { firstName, lastName, email, password, gender, error, success } = values;
20
21   const init = (userId) => {
22     // console.log(userId);
23     read(userId, token).then(data => {
24       if (data.error) {
25         setValues({ ...values, error: true });
26       }
27       else {
28         setValues({
29           ...values,
30           firstName: data.firstName,
31           lastName: data.lastName,
32           email: data.email,
33           gender: data.gender,
34         });
35       }
36     });
37   }
38 }
```

User model backend



log in frontend

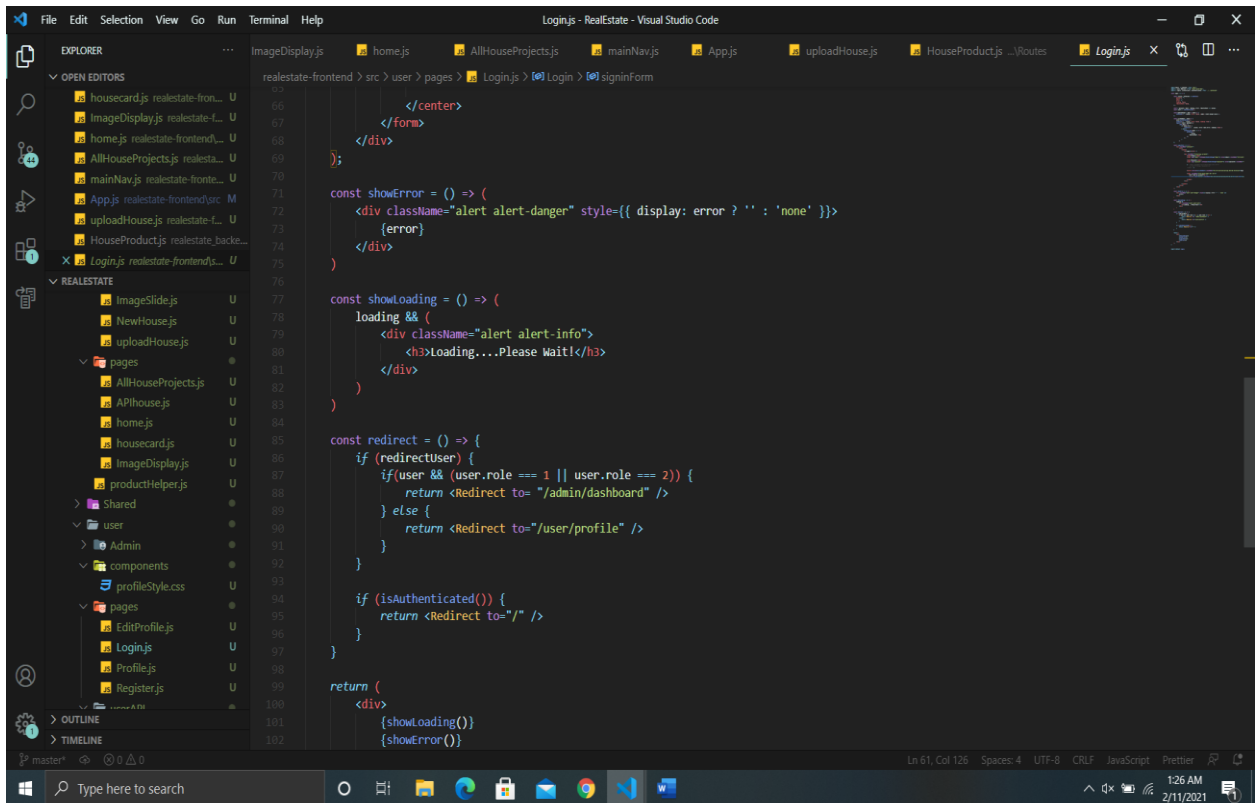
```
const { password, email, loading, error, redirectUser } = values;
const { user } = isAuthenticated();

const handleChange = name => event => {
  setValues({ ...values, error: false, [name]: event.target.value });
};

const clickSubmit = event => {
  event.preventDefault();
  setValues({ ...values, error: false, loading: true });
  signin({ email, password });
  .then(data => {
    if (data.error) {
      setValues({ ...values, error: data.error, loading: false });
    } else {
      authenticate(data, () => {
        setValues({
          ...values,
          redirectUser: true
        });
      });
    }
  });
});

const signInForm = () => {
  <div className="container">
    <form>
      <center>
        <h2>Login</h2></center>
        <div className="form-group col-md-6">
          <label>E-mail</label>
          <input type="email" onChange={handleChange("email")} value={email} className="form-control" />
          <label>Password</label>
```

login frontend



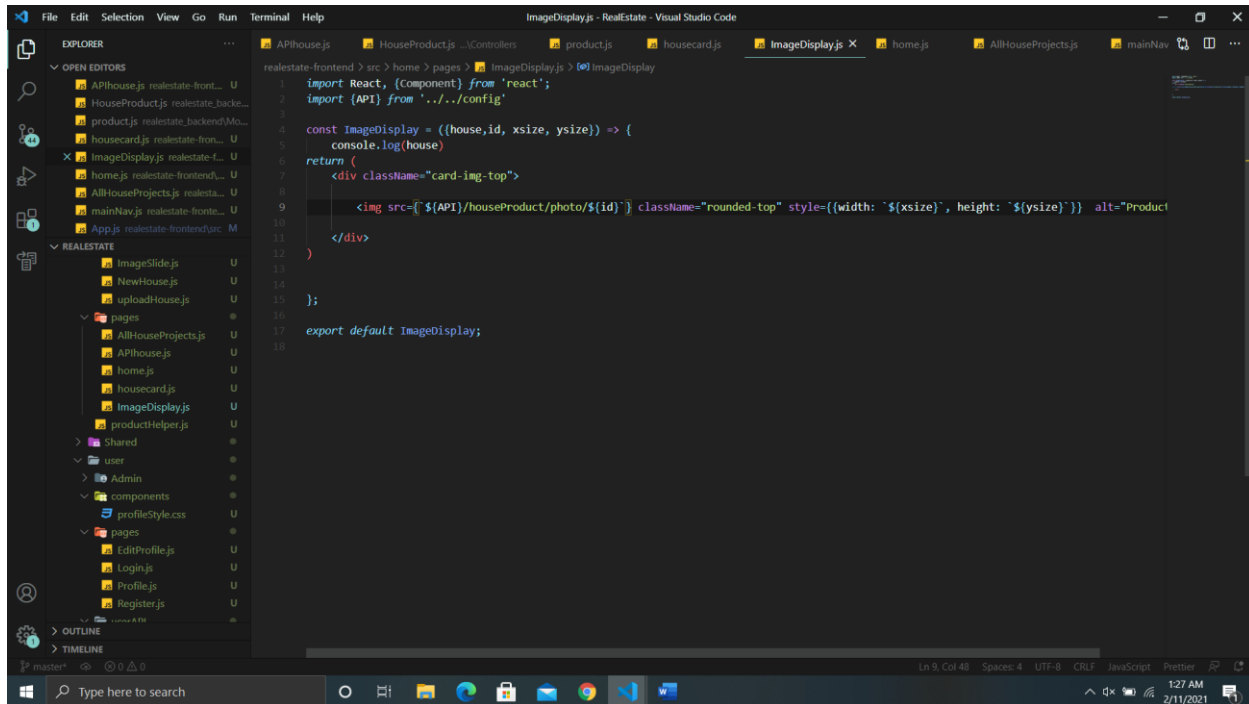
The screenshot shows the Visual Studio Code editor with the file `Login.js` open. The code is written in JavaScript and uses JSX for rendering. It includes functions for showing error messages, loading states, and handling user authentication and redirection. The Explorer sidebar on the left shows the project structure, including folders like `pages` and `components`. The status bar at the bottom indicates the current cursor position is at line 61, column 126.

```
File Edit Selection View Go Run Terminal Help
Login.js - RealEstate - Visual Studio Code

realstate-frontend > src > user > pages > Login.js > Login > signInForm

66         </center>
67       </form>
68     </div>
69   );
70
71   const showError = () => {
72     <div className="alert alert-danger" style={{ display: error ? '' : 'none' }}>
73       {error}
74     </div>
75   }
76
77   const showLoading = () => {
78     loading && (
79       <div className="alert alert-info">
80         <h3>Loading....Please Wait!</h3>
81       </div>
82     )
83   }
84
85   const redirect = () => {
86     if (redirectUser) {
87       if (user && (user.role === 1 || user.role === 2)) {
88         return <Redirect to="/admin/dashboard" />
89       } else {
90         return <Redirect to="/user/profile" />
91       }
92     }
93
94     if (isAuthenticated()) {
95       return <Redirect to="/" />
96     }
97   }
98
99   return (
100     <div>
101       {showLoading()}
102       {showError()}
103     </div>
104   );
105 }
106
107 export default Login;
```


image display of house card view file in frontend



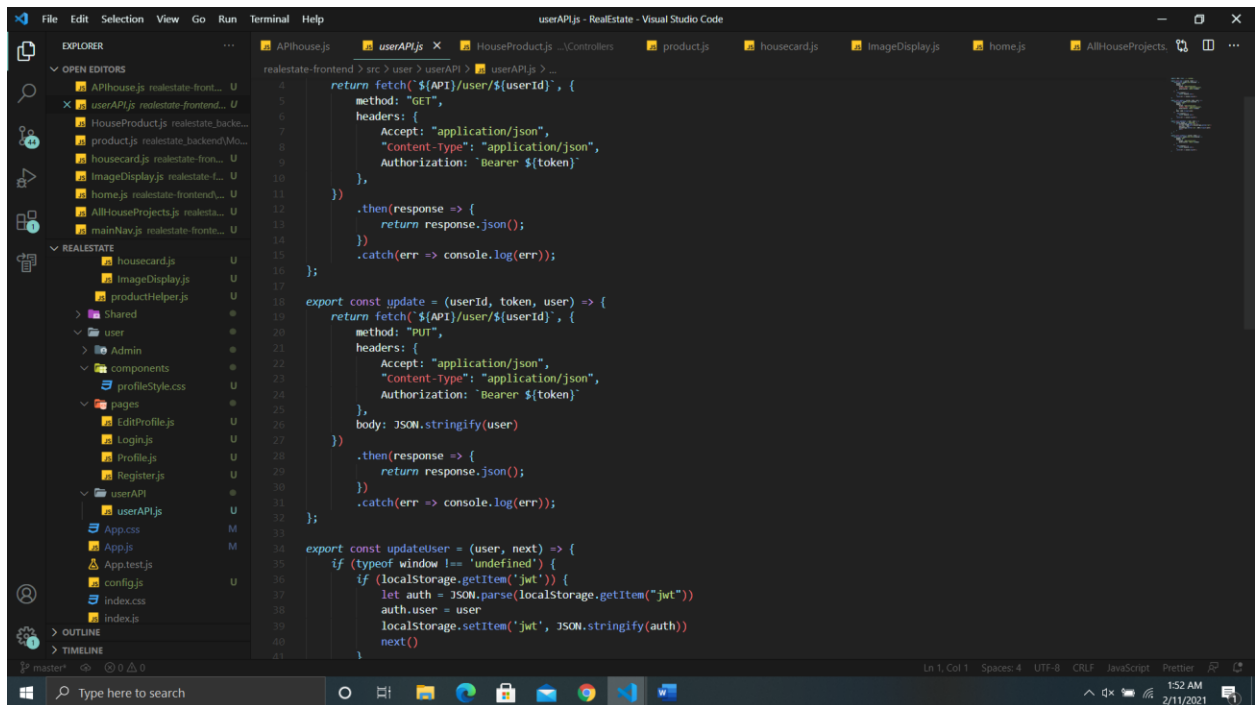
The screenshot shows the Visual Studio Code editor with the 'ImageDisplay.js' file open. The file is located in the 'pages' directory of the 'realstate-frontend' project. The code defines a functional component 'ImageDisplay' that takes 'house', 'id', 'xsize', and 'ysize' as props. It logs the 'house' object to the console and returns a JSX element. The JSX element consists of a 'div' with the class 'card-img-top' containing an 'img' tag. The 'img' tag's 'src' is constructed as `\${API}/houseProduct/photo/\${id}`, and its 'style' is an object with 'width' and 'height' properties set to 'xsize' and 'ysize' respectively. The 'alt' text is 'Product'. The component is exported as the default export.

```
1 import React, {Component} from 'react';
2 import {API} from '../../config'
3
4 const ImageDisplay = ({house,id, xsize, ysize}) => {
5   console.log(house)
6   return (
7     <div className="card-img-top">
8       <img src={` ${API}/houseProduct/photo/${id}`} className="rounded-top" style={{width: `${xsize}`, height: `${ysize}`}} alt="Product
9     </div>
10  )
11 }
12
13
14
15 };
16
17 export default ImageDisplay;
18
```

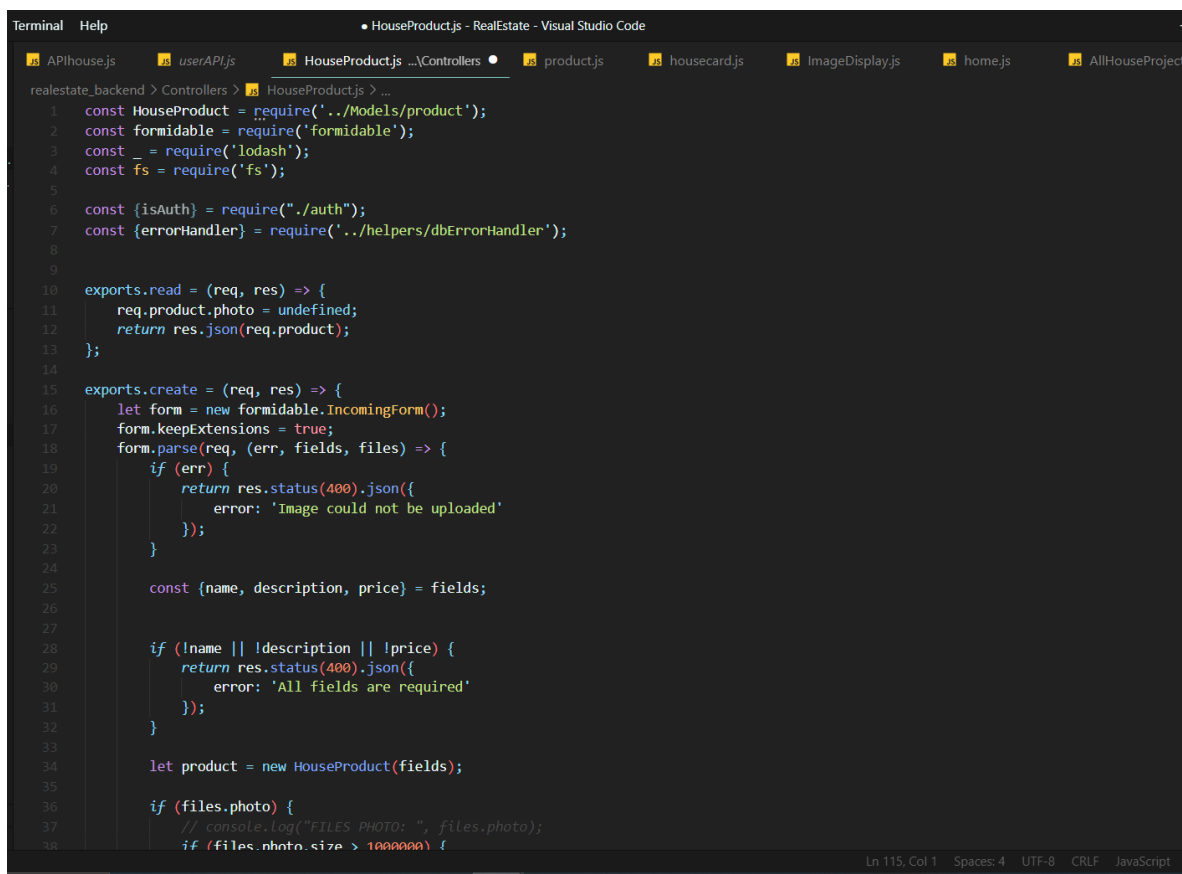
3. At least one implemented RESTful web service (implemented code).

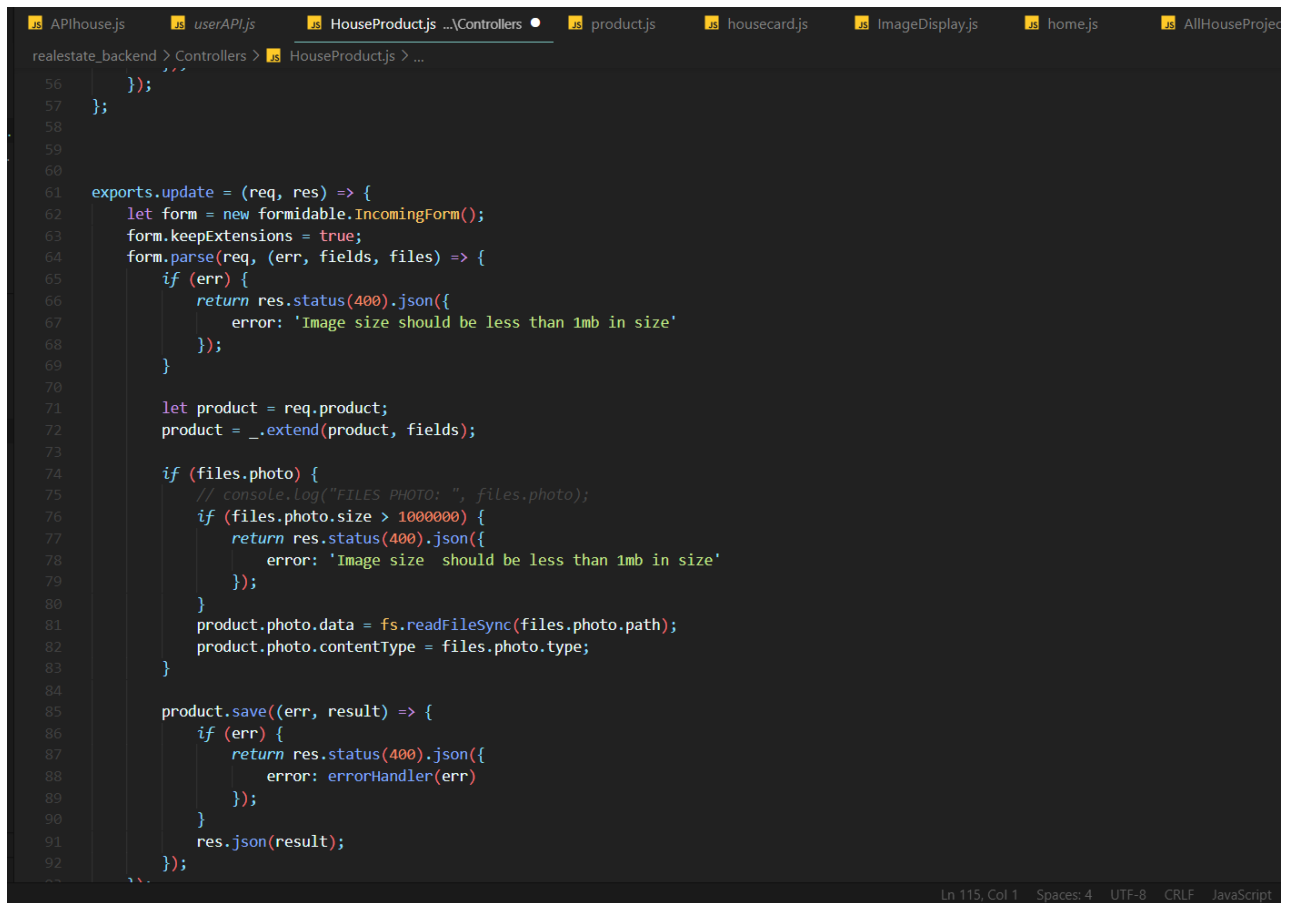
```
realstate-frontend > src > home > pages > APIhouse.js > getProductsForHome

1  import React from "react";
2  import { API } from "../../config";
3
4  export const createProduct = (product, token, userId) => {
5    console.log(product);
6    return fetch(`${API}/HouseProduct/create/${userId}`, {
7      method: 'POST',
8      headers: {
9        Accept: 'application/json',
10       Authorization: `Bearer ${token}`
11     },
12     body: product
13   }).then(response => {
14     return response.json();
15     console.log(response.json());
16   })
17   .catch(err => {
18     console.log('error in product creation',err);
19     alert(err);
20   });
21 };
22
23 export const getProductsForHome = (sortBy,limit) => {
24   return fetch(`${API}/houseProducts/withFilter?sortBy=${sortBy}&order=desc&limit=${limit}`, {
25     method: "GET"
26   })
27   .then(response => {
28     return response.json();
29   })
30   .catch(err => console.log(err));
31 };
32
33
34
35
36 // export const getProductsForHome = (sortBy,limit) => {
37 //   return fetch(`${process.env.REACT_APP_APIURL}/HouseProduct/product/`, {
38 //     method: "GET"
39 //   })
40 //   .then(response => {
41 //     return response.json();
42 //   })
43 //   .catch(err => console.log(err));
44 // };
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```



- At least one Mongo query for a specific Mongo collection (implemented code).

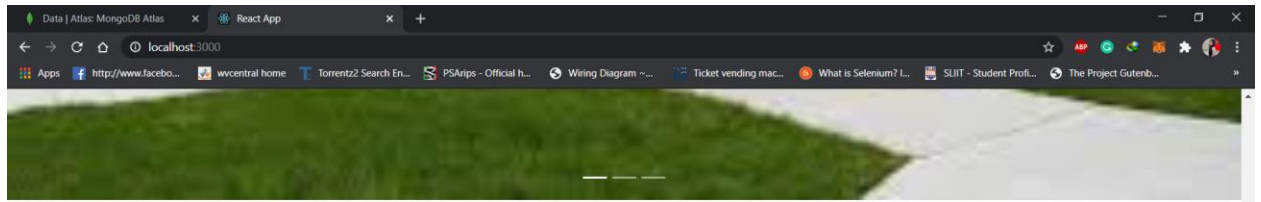




```
56     });
57   };
58
59
60
61   exports.update = (req, res) => {
62     let form = new formidable.IncomingForm();
63     form.keepExtensions = true;
64     form.parse(req, (err, fields, files) => {
65       if (err) {
66         return res.status(400).json({
67           error: 'Image size should be less than 1mb in size'
68         });
69       }
70
71       let product = req.product;
72       product = _.extend(product, fields);
73
74       if (files.photo) {
75         // console.log("FILES PHOTO: ", files.photo);
76         if (files.photo.size > 1000000) {
77           return res.status(400).json({
78             error: 'Image size should be less than 1mb in size'
79           });
80         }
81         product.photo.data = fs.readFileSync(files.photo.path);
82         product.photo.contentType = files.photo.type;
83       }
84
85       product.save((err, result) => {
86         if (err) {
87           return res.status(400).json({
88             error: errorHandler(err)
89           });
90         }
91         res.json(result);
92       });
93     });
94   }
95 }
```

5. At least one unit test using JEST to evaluate a core logic or a service (implemented code).

6. 6. A screenshot of the home page of the running application on localhost



Houses



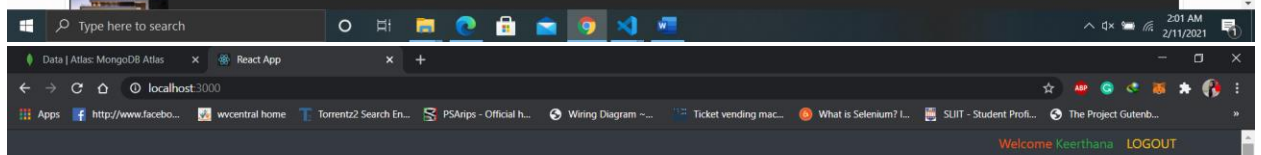
Name : Home With Swimming Pool

Product ID : 602431d7d6ba8a3240b26f2f

Price : \$ 45000000

Description : Amazing quality home in the terrific Parkland Heights Estate! If you are looking for a great home at a very affordable price then look no further as this home will tick all the boxes! This well presented 4 x 2 home has a lovely homely feel throughout and is ideal for anyone looking for a well designed home with all the comforts.

[Contact Us](#)



[Profile](#)

[Home](#)



