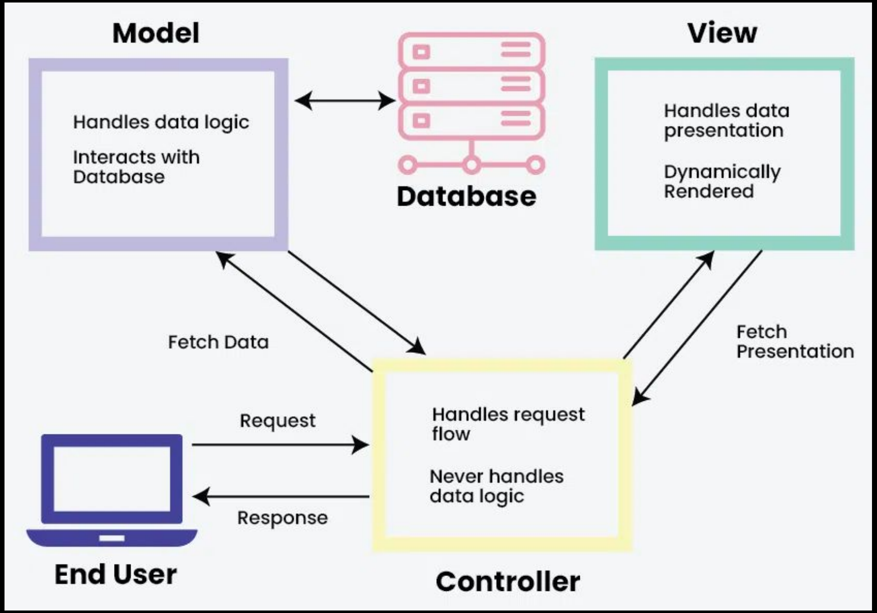
13. Model View Controller

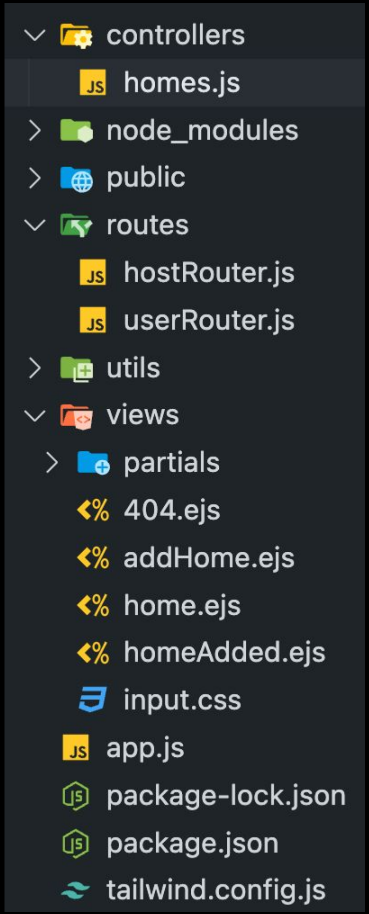
1. Separation of Concerns
2. Adding First Controller
3. Adding All Controllers
4. Adding 404 Controller
5. Adding Models
6. Writing data to Files
7. Nodemon causing problems
8. Reading data from Files

13.1 Separation of Concerns

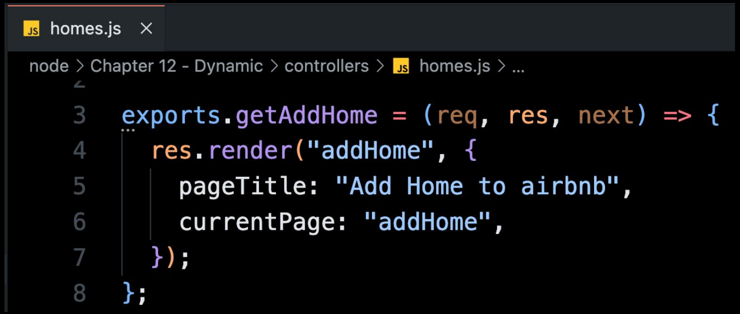
- **MVC stands for Model-View-Controller**: A software architectural pattern for developing user interfaces.
- **Model**: Manages the data and business logic of the application.
- **View**: Handles the display and presentation of data to the user.
- **Controller**: Processes user input, interacts with the Model, and updates the View accordingly.
- Routes are a part of Controllers.
- **Purpose**: MVC separates concerns within an application, making it easier to manage and scale.



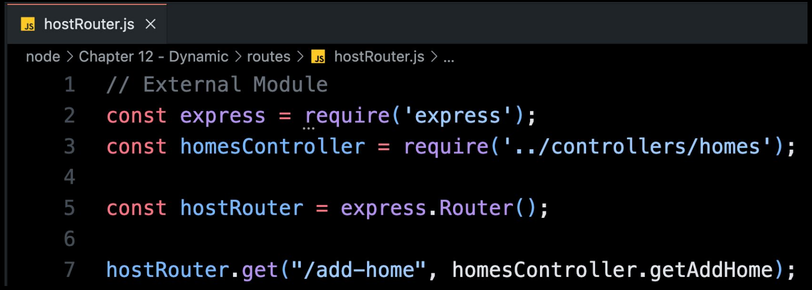
13.2 Adding First Controller



- controllers
  - homes.js
- node_modules
- public
- routes
  - hostRouter.js
  - userRouter.js
- utils
- views
  - partials
  - 404.ejs
  - addHome.ejs
  - home.ejs
  - homeAdded.ejs
  - input.css
- app.js
- package-lock.json
- package.json
- tailwind.config.js

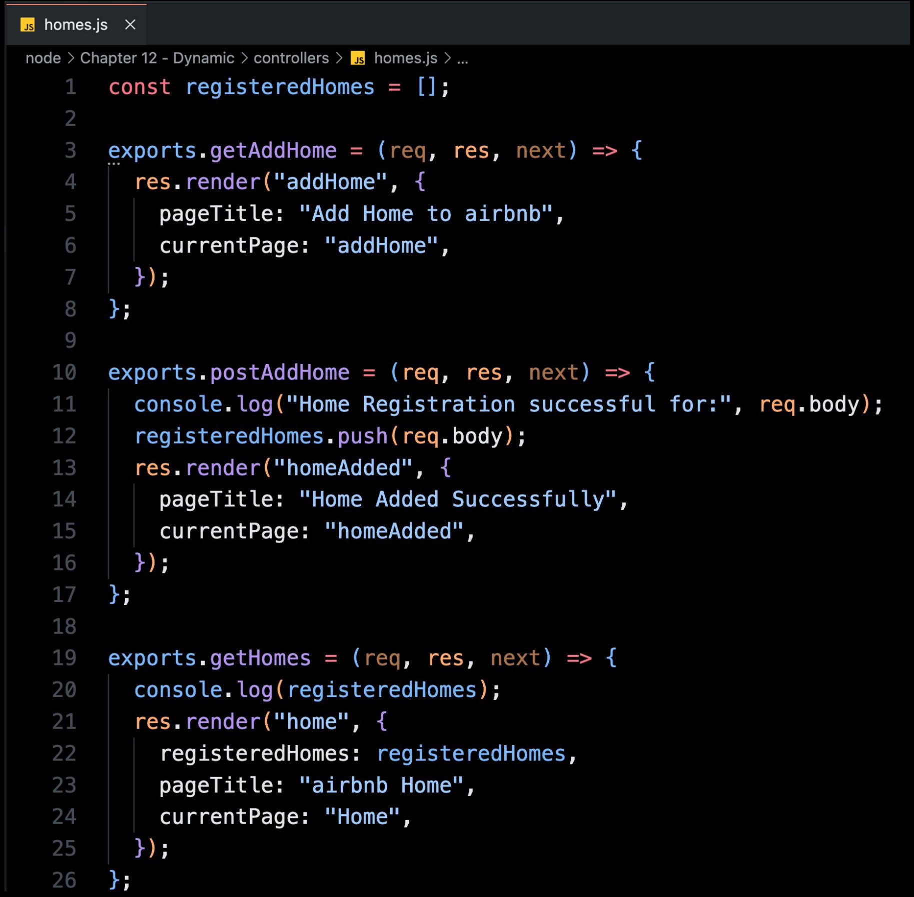


```
exports.getAddHome = (req, res, next) => {
  res.render("addHome", {
    pageTitle: "Add Home to airbnb",
    currentPage: "addHome",
  });
};
```



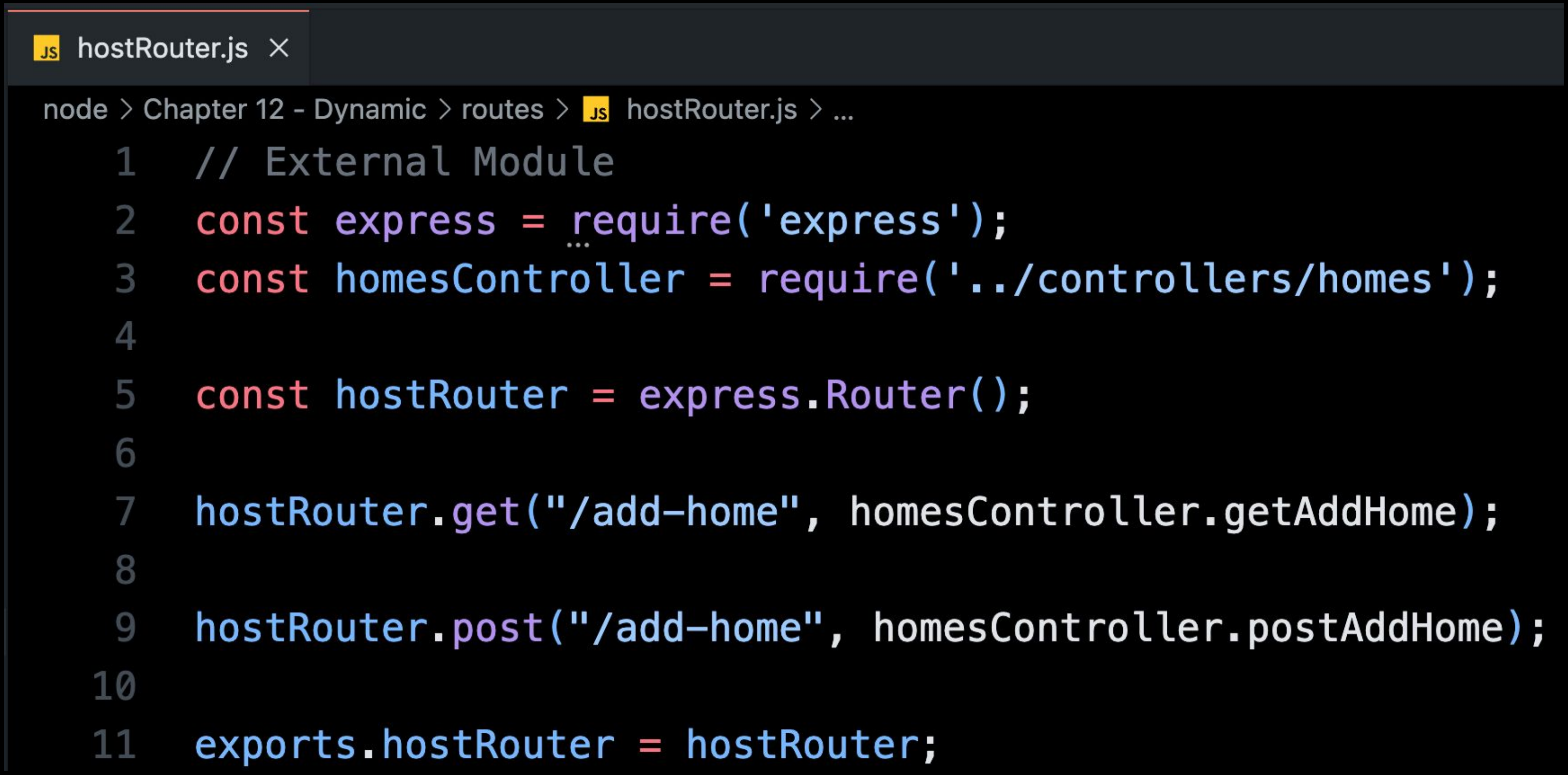
```
// External Module
const express = require('express');
const homesController = require('../controllers/homes');

const hostRouter = express.Router();

hostRouter.get("/add-home", homesController.getAddHome);
```

13.3 Adding All Controllers



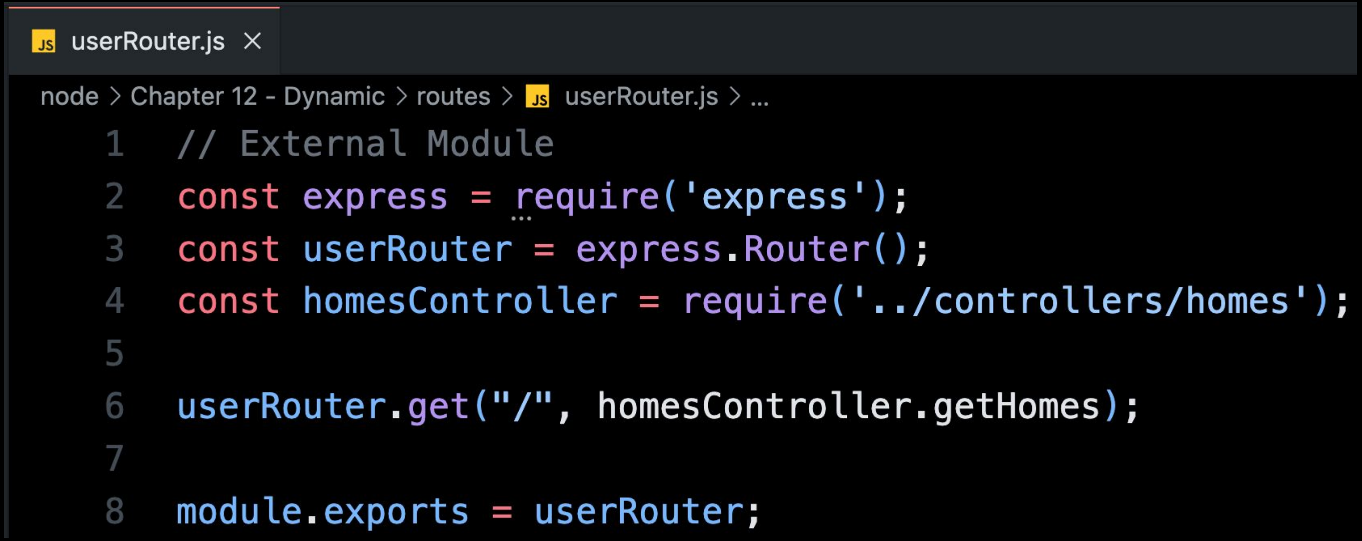
```javascript
const registeredHomes = [];

exports.getAddHome = (req, res, next) => {
  res.render("addHome", {
    pageTitle: "Add Home to airbnb",
    currentPage: "addHome",
  });
};

exports.postAddHome = (req, res, next) => {
  console.log("Home Registration successful for:", req.body);
  registeredHomes.push(req.body);
  res.render("homeAdded", {
    pageTitle: "Home Added Successfully",
    currentPage: "homeAdded",
  });
};

exports.getHomes = (req, res, next) => {
  console.log(registeredHomes);
  res.render("home", {
    registeredHomes: registeredHomes,
    pageTitle: "airbnb Home",
    currentPage: "Home",
  });
};
```

13.3 Adding All Controllers



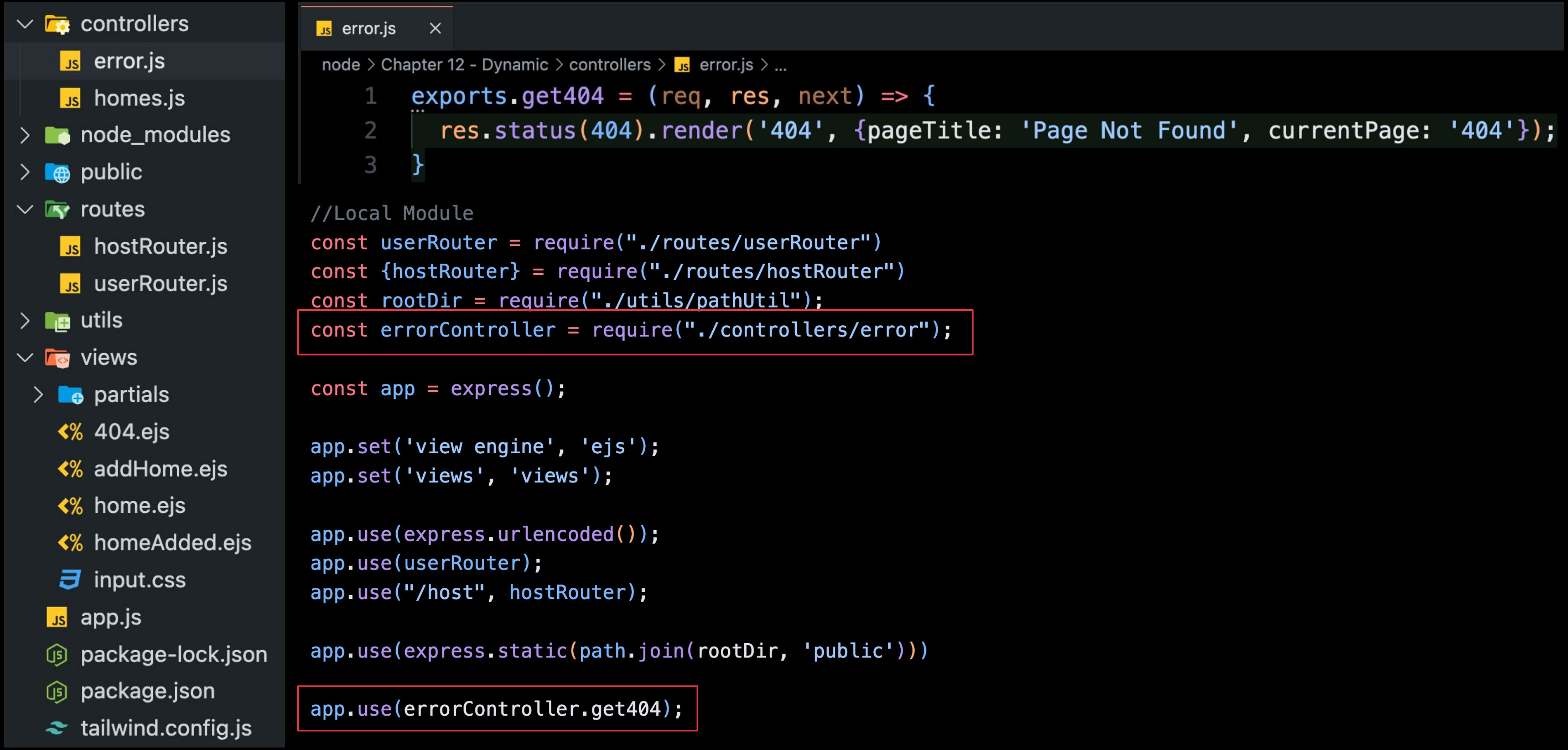
```javascript
// External Module
const express = require('express');
const homesController = require('../controllers/homes');

const hostRouter = express.Router();

hostRouter.get("/add-home", homesController.getAddHome);

hostRouter.post("/add-home", homesController.postAddHome);

exports.hostRouter = hostRouter;
```

13.3 Adding All Controllers



```javascript
// External Module
const express = require('express');
const userRouter = express.Router();
const homesController = require('../controllers/homes');

userRouter.get("/", homesController.getHomes);

module.exports = userRouter;
```

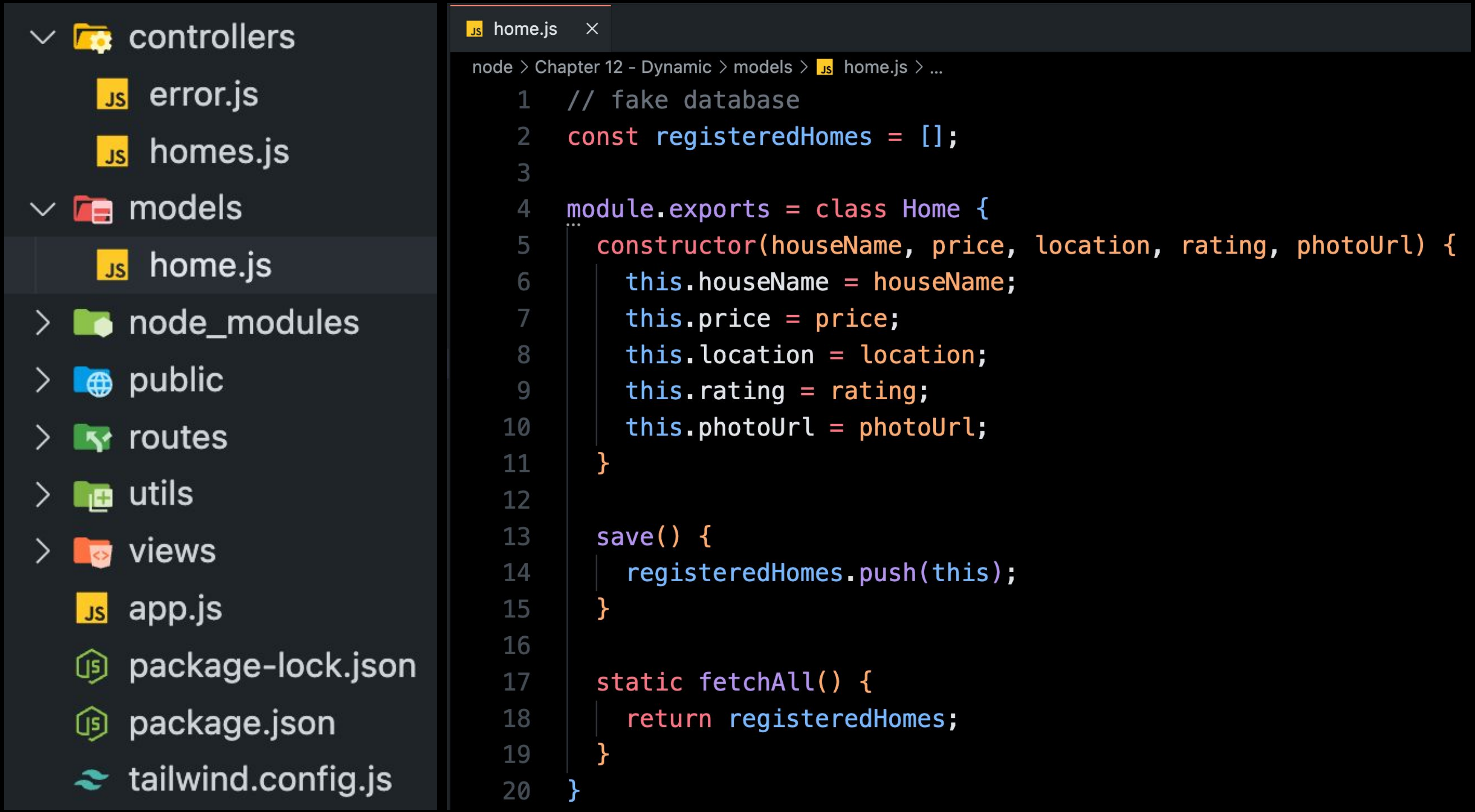
13.4 Adding 404 Controller



```
exports.get404 = (req, res, next) => {
  res.status(404).render('404', {pageTitle: 'Page Not Found', currentPage: '404'});
}
```

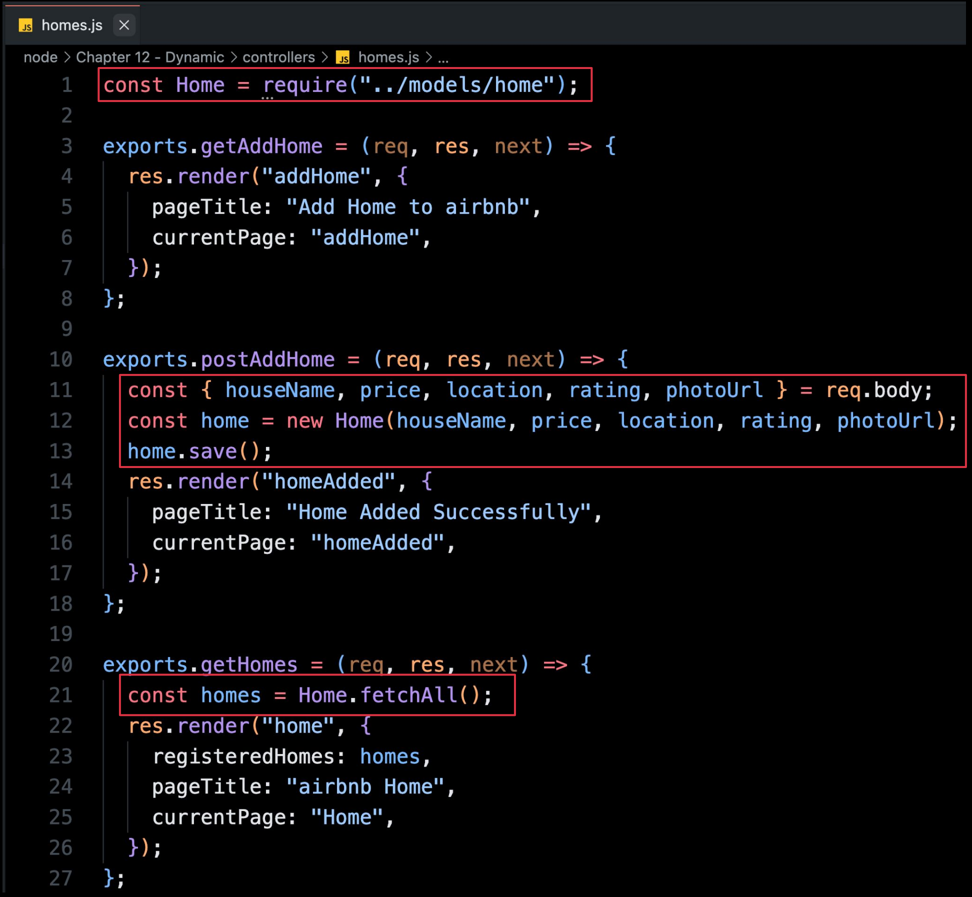
```
//Local Module
const userRouter = require("./routes/userRouter")
const {hostRouter} = require("./routes/hostRouter")
const rootDir = require("./utils/pathUtil");
const errorController = require("./controllers/error");

const app = express();

app.set('view engine', 'ejs');
app.set('views', 'views');

app.use(express.urlencoded());
app.use(userRouter);
app.use("/host", hostRouter);

app.use(express.static(path.join(rootDir, 'public')))

app.use(errorController.get404);
```

13.5 Adding Models



```javascript
// fake database
const registeredHomes = [];

module.exports = class Home {
  constructor(houseName, price, location, rating, photoUrl) {
    this.houseName = houseName;
    this.price = price;
    this.location = location;
    this.rating = rating;
    this.photoUrl = photoUrl;
  }

  save() {
    registeredHomes.push(this);
  }

  static fetchAll() {
    return registeredHomes;
  }
}
```

13.5 Adding Models

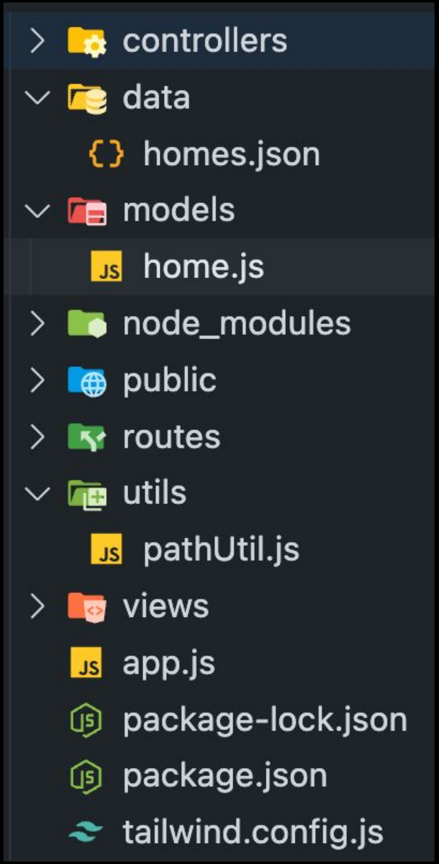


homes.js

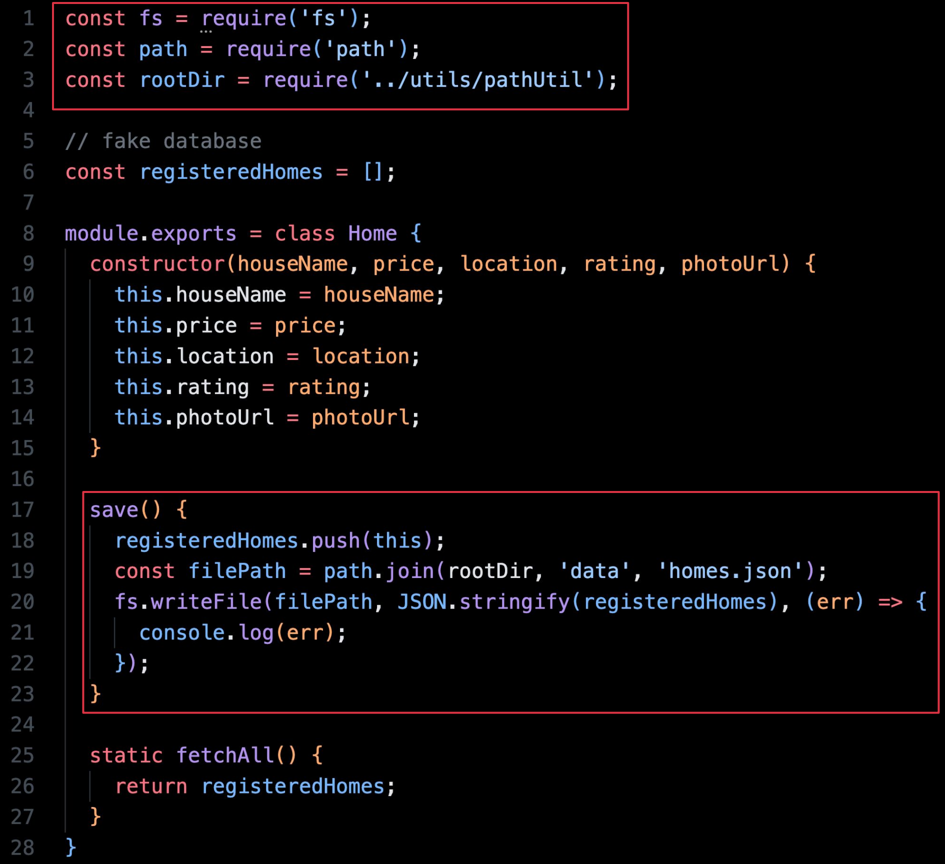
node > Chapter 12 - Dynamic > controllers > homes.js > ...

```
const Home = require("../models/home");

exports.getAddHome = (req, res, next) => {
  res.render("addHome", {
    pageTitle: "Add Home to airbnb",
    currentPage: "addHome",
  });
};

exports.postAddHome = (req, res, next) => {
  const { houseName, price, location, rating, photoUrl } = req.body;
  const home = new Home(houseName, price, location, rating, photoUrl);
  home.save();
  res.render("homeAdded", {
    pageTitle: "Home Added Successfully",
    currentPage: "homeAdded",
  });
};

exports.getHomes = (req, res, next) => {
  const homes = Home.fetchAll();
  res.render("home", {
    registeredHomes: homes,
    pageTitle: "airbnb Home",
    currentPage: "Home",
  });
};
```

13.6 Writing data to Files

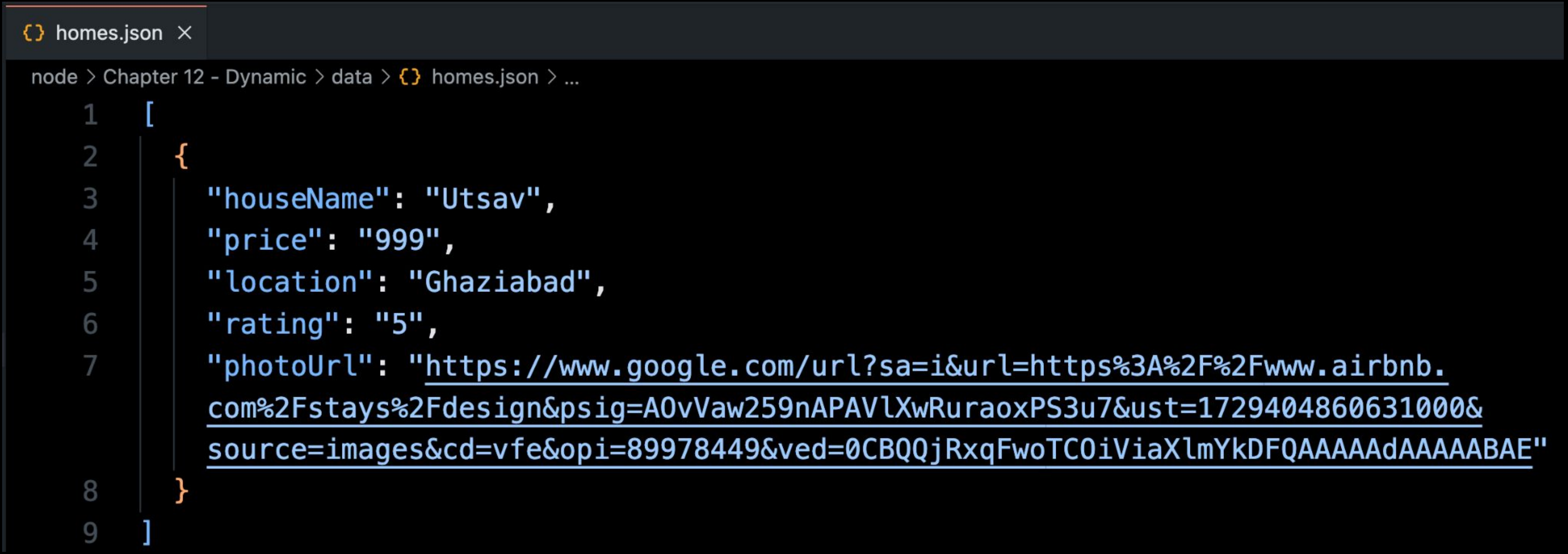


- controllers
- data
  - homes.json
- models
  - home.js
- node_modules
- public
- routes
- utils
  - pathUtil.js
- views
- app.js
- package-lock.json
- package.json
- tailwind.config.js



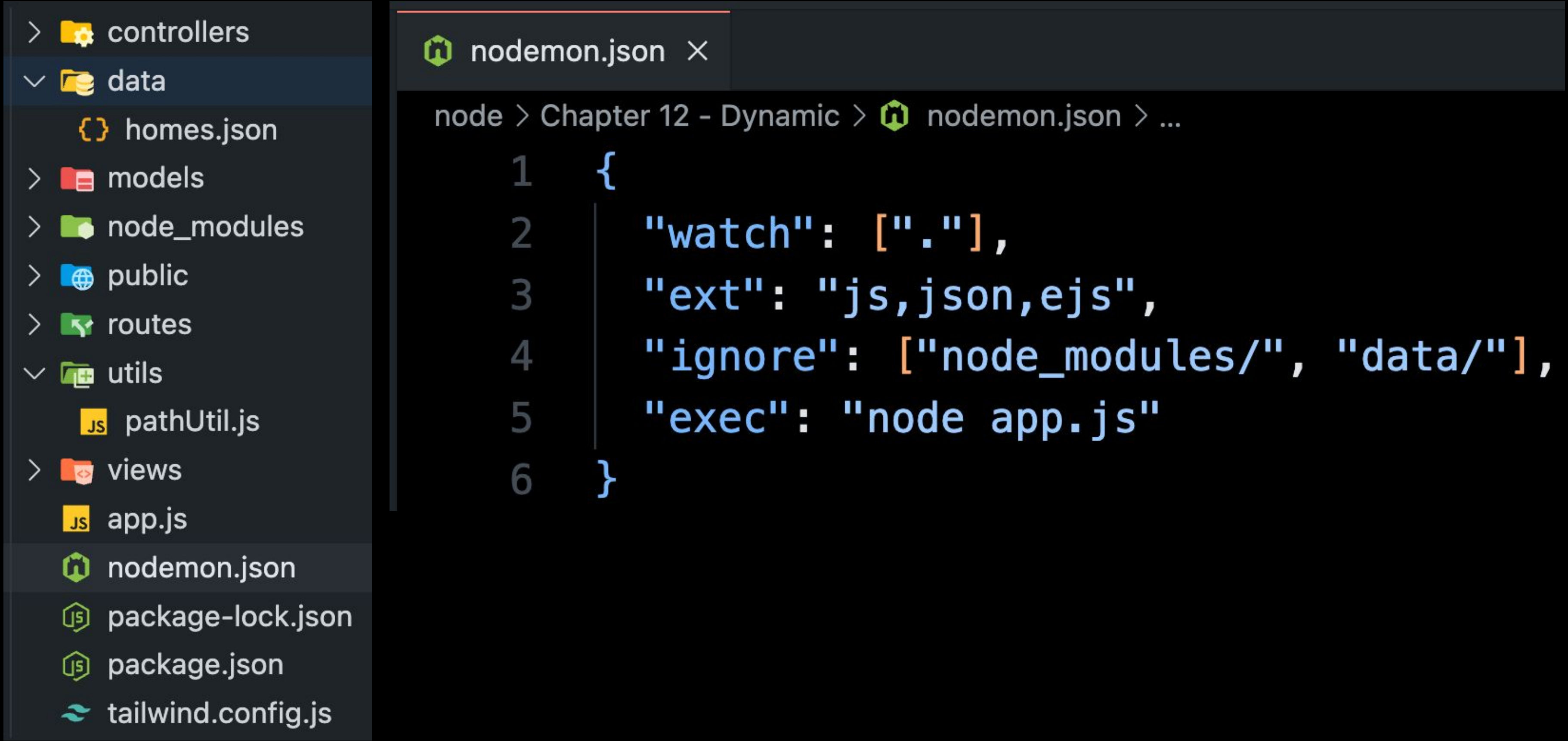
```
const fs = require('fs');
const path = require('path');
const rootDir = require('../utils/pathUtil');

// fake database
const registeredHomes = [];

module.exports = class Home {
  constructor(houseName, price, location, rating, photoUrl) {
    this.houseName = houseName;
    this.price = price;
    this.location = location;
    this.rating = rating;
    this.photoUrl = photoUrl;
  }

  save() {
    registeredHomes.push(this);
    const filePath = path.join(rootDir, 'data', 'homes.json');
    fs.writeFile(filePath, JSON.stringify(registeredHomes), (err) => {
      console.log(err);
    });
  }

  static fetchAll() {
    return registeredHomes;
  }
}
```

13.6 Writing data to Files



```json
[
  {
    "houseName": "Utsav",
    "price": "999",
    "location": "Ghaziabad",
    "rating": "5",
    "photoUrl": "https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.airbnb.com%2Fstays%2Fdesign&psig=AOvVaw259nAPAVlXwRuraoxPS3u7&ust=1729404860631000&source=images&cd=vfe&opi=89978449&ved=0CBQQjRxqFwoTCOiViaXlmYkDFQAAAAdAAAAABAE"
  }
]
```

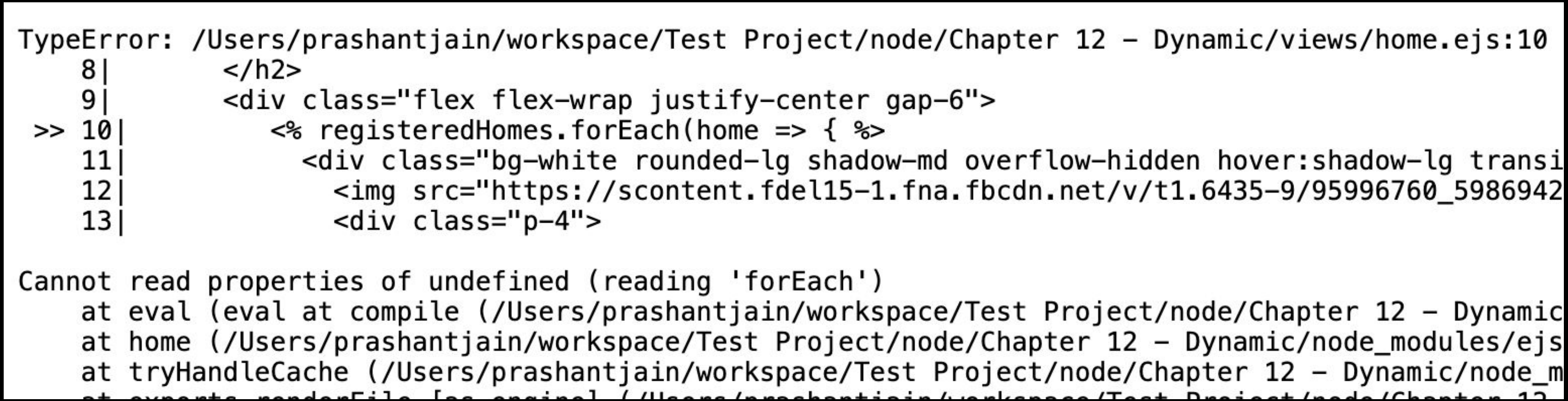
13.7 Nodemon causing problems



```json
{
    "watch": ["."],
    "ext": "js,json,ejs",
    "ignore": ["node_modules/", "data/"],
    "exec": "node app.js"
}
```

13.8 Reading data from Files

```
static fetchAll() {
  const filePath = path.join(rootDir, 'data', 'homes.json');
  const fileContent = fs.readFile(filePath, (err, data) => {
    if (err) {
      return [];
    }
    return JSON.parse(data);
  });
}
```



```
TypeError: /Users/prashantjain/workspace/Test Project/node/Chapter 12 - Dynamic/views/home.ejs:10
    8|         </h2>
    9|         <div class="flex flex-wrap justify-center gap-6">
 >> 10|           <% registeredHomes.forEach(home => { %>
    11|             <div class="bg-white rounded-lg shadow-md overflow-hidden hover:shadow-lg transi
    12|               <img src="https://scontent.fdel15-1.fna.fbcdn.net/v/t1.6435-9/95996760_5986942
    13|               <div class="p-4">

Cannot read properties of undefined (reading 'forEach')
    at eval (eval at compile (/Users/prashantjain/workspace/Test Project/node/Chapter 12 - Dynamic
    at home (/Users/prashantjain/workspace/Test Project/node/Chapter 12 - Dynamic/node_modules/ejs
    at tryHandleCache (/Users/prashantjain/workspace/Test Project/node/Chapter 12 - Dynamic/node_m
    at exports.renderFile [as engine] (/Users/prashantjain/workspace/Test Project/node/Chapter 12
```

13.8 Reading data from Files

```
save() {
  this.fetchAll((registeredHomes) => {
    registeredHomes.push(this);
    const filePath = path.join(rootDir, 'data', 'homes.json');
    fs.writeFile(filePath, JSON.stringify(registeredHomes), (err) => {
      console.log(err);
    });
  });
}

static fetchAll(callback) {
  const filePath = path.join(rootDir, 'data', 'homes.json');
  fs.readFile(filePath, (err, data) => {
    let homes = [];
    if (!err) {
      homes = JSON.parse(data);
    }
    callback(homes);
  });
}
```

```
exports.getHomes = (req, res, next) => {
  Home.fetchAll((homes) => {
    res.render("home", {
      registeredHomes: homes,
      pageTitle: "airbnb Home",
      currentPage: "Home",
    });
  });
};
```

Practise Milestone

Take your airbnb forward:

1. Structure the views folder into host & store and move the respective view files there.
2. Add more views to store like: home-list, home-detail, favourite-list, reserve, bookings. And to host view: edit-home, host-home-list
3. Improve the header with navigation to all pages.
4. Register all the new routes and add dummy views there.
5. Change controllers to store and host setup.
6. Add Edit and Delete buttons to the host-home-list view.
7. Keep the logic for Edit, Delete, Favourite pending.

Revision

1. Separation of Concerns
2. Adding First Controller
3. Adding All Controllers
4. Adding 404 Controller
5. Adding Models
6. Writing data to Files
7. Nodemon causing problems
8. Reading data from Files