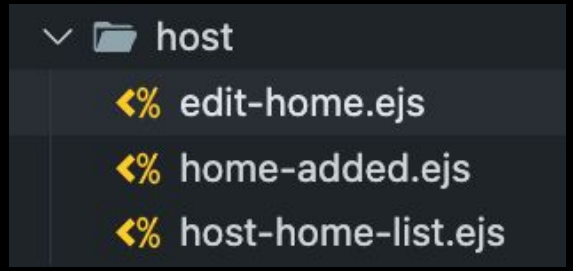
14.6 Edit-Home: Adding Feature Skeleton

1. Rename the add-home.ejs to edit-home.ejs
2. Fix it's usage in the add-home controller.
3. Change the path of edit button everywhere.
4. Add a new router for GET /edit-home/:home-id
5. Add a new controller method in host controller, optionally logging query param of editing=true, while passing editing flag to view.

14.6 Edit-Home: Adding Feature Skeleton

1.



2.

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

14.7 Edit-Home: Showing Existing Data

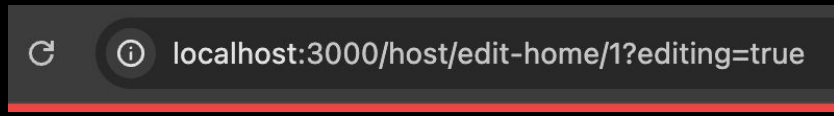
3.

```
<%- include('../partials/head') %>
  <body>
    <%- include('../partials/nav') %>
    <main class="container mx-auto bg-white shadow-lg rounded-lg p-8 mt-10 max-w-6xl">
      <div class="flex flex-wrap justify-center gap-6">
                </div>
                <a href="/host/edit-home/<%= home.id %>?editing=true" class="bg-blue-500 p-2">Edit</a>
                <button class="bg-red-500 p-2">Delete</button>
              </div>
            </div>
          <% }) %>
      </div>
    </main>
  </body>
</html>
```

14.6 Edit-Home: Adding Feature Skeleton

4.
```
hostRouter.get("/edit-home/:homeId", hostController.getEditHome);
```

5.
```
exports.getEditHome = (req, res, next) => {
  const editing = req.query.editing === "true";
  const homeId = req.params.homeId;
  console.log(editing, homeId);
  res.render("host/edit-home", {
    pageTitle: "Edit Home",
    currentPage: "editHome",
    editing: editing,
    homeId: homeId,
  });
};
```




```
Server running on address http://localhost:3000
false 1
true 1
```

14.7 Edit-Home: Showing Existing Data

1. Change the controller to now fetch the home details, if not found redirect to /host/host-home-list otherwise passing the data to view.
2. Change the edit-home.ejs to show dynamic content based on values:
    a. Different submit path.
    b. Different button text.
    c. Pre-filled values if present.

14.7 Edit-Home: Showing Existing Data

1.
```
exports.getEditHome = (req, res, next) => {
  const editing = req.query.editing === "true";
  const homeId = req.params.homeId;
  Home.findById(homeId, (home) => {
    if (!home) {
      return res.redirect("/host/host-home-list");
    }
    res.render("host/edit-home", {
      pageTitle: "Edit Home",
      currentPage: "editHome",
      editing: editing,
      home: home,
    });
  });
};
```

14.7 Edit-Home: Showing Existing Data

2.

```
<%- include('../partials/head') %>
  </head>
  <body>
    <%- include('../partials/nav') %>
    <main class="container mx-auto mt-8 p-8 bg-white rounded-lg shadow-md">
      <h1 class="text-3xl font-bold mb-6 text-center text-gray-800">Register Your Home on AirBnB</h1>
      <form action="/host/<%= editing ? 'edit-home' : 'add-home' %>" method="POST" class="max-w-md mx-auto">
        <input
          type="text"
          name="houseName"
          value="<%= home ? home.houseName : '' %>"
          placeholder="Enter your House Name"
          class="w-full px-4 py-2 mb-4 border rounded-md focus:outline-none focus:ring-2 focus:ring-red-500"/>
        <input
          type="text"
          name="price"
          value="<%= home ? home.price : '' %>"
          placeholder="Price Per Night"
          class="w-full px-4 py-2 mb-4 border rounded-md focus:outline-none focus:ring-2 focus:ring-red-500"/>
        <input
          type="text"
          name="location"
          value="<%= home ? home.location : '' %>"
          placeholder="Location"
          class="w-full px-4 py-2 mb-4 border rounded-md focus:outline-none focus:ring-2 focus:ring-red-500"/>
        <input
          type="text"
          name="rating"
          value="<%= home ? home.rating : '' %>"
          placeholder="Rating"
          class="w-full px-4 py-2 mb-4 border rounded-md focus:outline-none focus:ring-2 focus:ring-red-500"/>
        <input
          type="text"
          name="photoUrl"
          value="<%= home ? home.photoUrl : '' %>"
          placeholder="Enter Photo URL"
          class="w-full px-4 py-2 mb-4 border rounded-md focus:outline-none focus:ring-2 focus:ring-red-500"/>
        <input type="submit" value="<%= editing ? 'Update' : 'Add' %> Home" class="w-full bg-red-500 text-white py-;
        transition duration-300"/>
      </form>
    </main>
  </body>
</html>
```

14.8 Edit-Home: Handing Edit Request

1. Add a router for POST /edit-home
2. Add a controller for /edit-home, which creates a home model object and saves it, before redirecting to /host/host-home-list
3. Add hidden id input field in the edit-home.ejs to get the id as part of POST request.
4. Change the save method in Model to handle creation of new as well as updation of existing Home.

14.8 Edit-Home: Handing Edit Request

1.
```
hostRouter.post("/edit-home", hostController.postEditHome);
```

2.
```
exports.postEditHome = (req, res, next) => {
  const { id, houseName, price, location, rating, photoUrl } = req.body;
  const home = new Home(houseName, price, location, rating, photoUrl);
  home.id = id;
  home.save();
  res.redirect("/host/host-home-list");
};
```

3.
```
<%- include('../partials/head') %>
  <body>
    <%- include('../partials/nav') %>
    <main class="container mx-auto mt-8 p-8 bg-white rounded-lg shadow-md">
      <h1 class="text-3xl font-bold mb-6 text-center text-gray-800">Register
      <form action="/host/<%= editing ? 'edit-home' : 'add-home' %>" method="P
        <input type="hidden" name="id" value="<%= home ? home.id : '' %>">
        <input
          type="text"
          name="houseName"
```

14.8 Edit-Home: Handing Edit Request

4.

```
module.exports = class Home {

  save() {
    Home.fetchAll((registeredHomes) => {
      console.log(registeredHomes, this);
      if (this.id) {
        registeredHomes = registeredHomes.map((home) => {
          console.log(home.id, this.id);
          if (home.id === this.id) {
            return this;
          }
          return home;
        });
      } else {
        console.log("New Home");
        this.id = Math.random().toString();
        registeredHomes.push(this);
      }
      const homeDataPath = path.join(rootDir, "data", "homes.json");
      fs.writeFile(homeDataPath, JSON.stringify(registeredHomes), (error) => {
        console.log("File Writing Concluded", error);
      });
    });
  }
```

14.9 Adding Delete Feature

1. Surround the delete button with a form that submits to path /host/delete-home/:homeId
2. Add a route in the host routes.
3. Add a static delete method to the Home model that takes an id and deletes the home.
4. Add a method in host controller to handle the request, delete the home and redirect to /host/host-homes-list page.
5. Pending: deleted homes might still be in favourites list.

14.9 Adding Delete Feature

1.
```
<a href="/host/edit-home/<%= home.id %>?editing=true" class="bg-blue-500 p-2 rounded
text-white hover:bg-blue-600">Edit</a>
<form action="/host/delete-home/<%= home.id %>" method="POST" class="inline">
  <button type="submit" class="bg-red-500 p-2 rounded text-white
  hover:bg-red-600">Delete</button>
</form>
```

2.
```
hostRouter.post("/delete-home/:homeId", hostController.postDeleteHome);
```

3.
```
static deleteById(id, callback) {
  this.fetchAll((homes) => {
    const updatedHomes = homes.filter((home) => home.id !== id);
    fs.writeFile(path.join(rootDir, "data", "homes.json"), JSON.stringify(updatedHomes), callback);
  });
}
```

14.9 Adding Delete Feature

4.
```
exports.postDeleteHome = (req, res, next) => {
  const homeId = req.params.homeId;
  Home.deleteById(homeId, (error) => {
    if (error) {
      console.log("Error Deleting Home", error);
    }
    res.redirect("/host/host-home-list");
  });
};
```

14.10 Removing Home from Favourites

1. Add a static method to the favourites model to delete the home.
2. Add a form to the favourites-list in each home to path /favourites/delete/:homeId
3. Add a route for POST delete-favourites in the store routes.
4. Add a method in store controller to use the model's static method and then redirect to favourites-list.
5. Use this model method in home-delete to make sure any deleted home is also removed from favourites.

14.10 Removing Home from Favourites

1.
```
static deleteFavourite(homeId, callback) {
  this.getFavourites((favourites) => {
    const updatedFavourites = favourites.filter((id) => id !== homeId);
    fs.writeFile(favouritesPath, JSON.stringify(updatedFavourites), callback);
  });
}
```

2.
```
<form action="/favourites/delete/<%= home.id %>" method="POST">
  <button type="submit" class="bg-red-500 p-2 rounded text-white
  hover:bg-red-600">Delete</button>
</form>
```

3.
```
storeRouter.post("/favourites/delete/:homeId", storeController.deleteFavourite);
```

14.10 Removing Home from Favourites

4.

```
exports.deleteFavourite = (req, res, next) => {
  const homeId = req.params.homeId;
  Favourites.deleteFavourite(homeId, (error) => {
    if (error) {
      console.log("Error Deleting Favourite", error);
    }
    res.redirect('/favourites');
  });
};
```

5.

```
static deleteById(id, callback) {
  this.fetchAll((homes) => {
    const updatedHomes = homes.filter((home) => home.id !== id);
    fs.writeFile(path.join(rootDir, "data", "homes.json"), JSON.stringify(updatedHomes), (error) => {
      Favourites.deleteFavourite(id, callback);
    });
  });
}
```