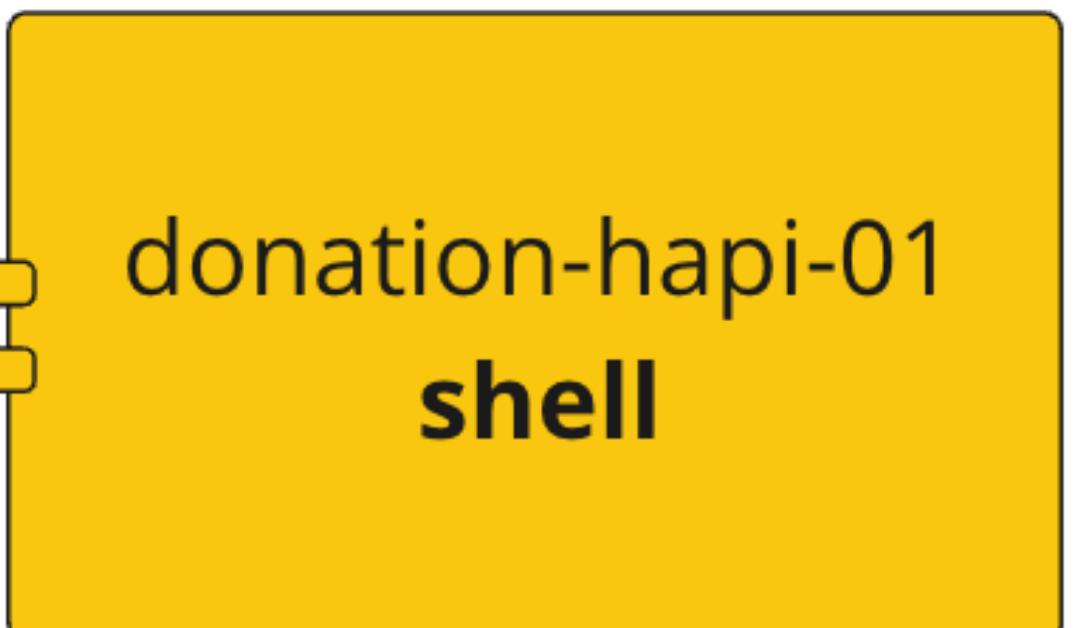
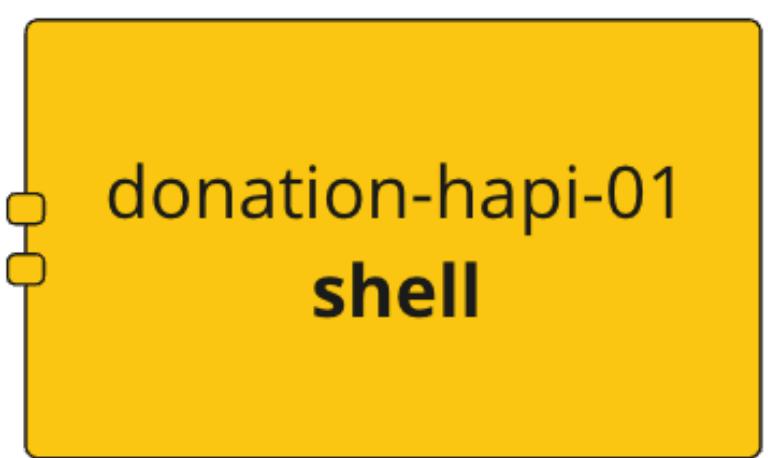


```
✓ public
  ↗ homer-1.png
  ↗ homer-2.jpeg
  # main.css
✓ src
  ✓ controllers
    JS accounts-controller.js
    JS donations-controller.js
  ✓ models
    ✓ mongo
      JS connect.js
      JS seed-data.js
      JS user-store.js
      JS user.js
    JS db.js
  ✓ views
    ✓ partials
      ~ heading.hbs
      ~ login-form.hbs
      ~ menu.hbs
      ~ signup-form.hbs
      ~ user-credentials.hbs
      ~ user-details.hbs
      ~ donate.hbs
      ~ layout.hbs
      ~ login.hbs
      ~ main.hbs
      ~ signup.hbs
    JS server.js
    JS web-routes.js
  .env
  .eslintrc.json
  .gitignore
  .prettierrc.json
  package-lock.json
  package.json
  README.md
```

# donation-hapi-01-shell



- Basic User System + Placeholder views



```
✓ src
  ✓ controllers
    JS accounts-controller.js
    JS donations-controller.js
```

```
✓ models
  ✓ mongo
    JS connect.js
    JS seed-data.js
    JS user-store.js
    JS user.js
  JS db.js
```

```
JS server.js
JS web-routes.js
```

```
✓ views
  ✓ partials
    ~ heading.hbs
    ~ login-form.hbs
    ~ menu.hbs
    ~ signup-form.hbs
    ~ user-credentials.hbs
    ~ user-details.hbs
    ~ donate.hbs
    ~ layout.hbs
    ~ login.hbs
    ~ main.hbs
    ~ signup.hbs
```

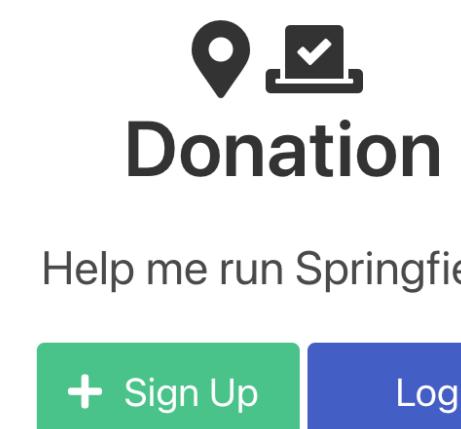


## Login to DONATION

Email

Password

Log In



## Signup for DONATION

Name

Email

Password

Create Account





```
✓ src
  ✓ controllers
    JS accounts-controller.js
    JS donations-controller.js
```

```
✓ models
  ✓ mongo
    JS connect.js
    JS seed-data.js
    JS user-store.js
    JS user.js
  JS db.js
```

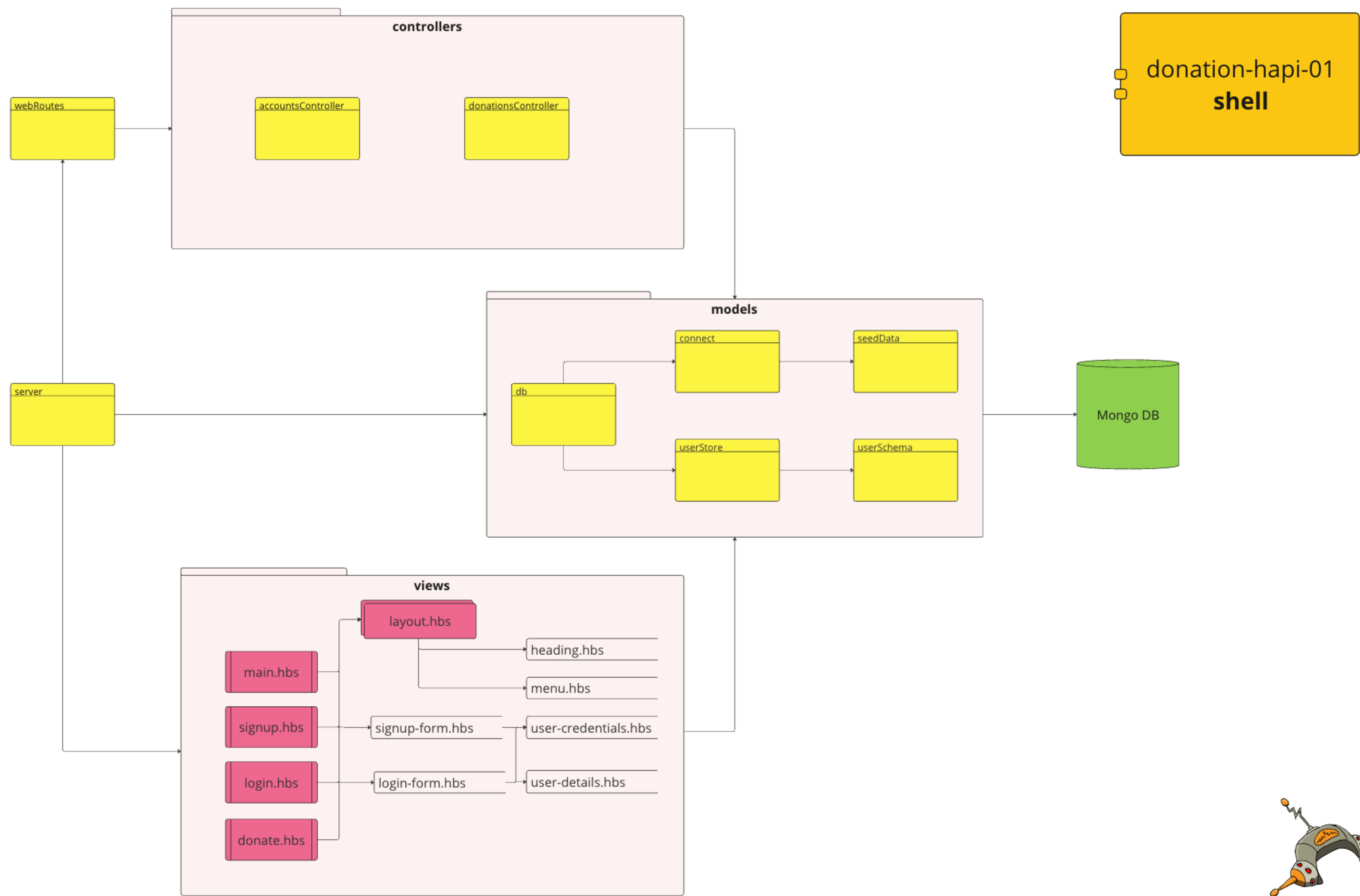
```
JS server.js
JS web-routes.js
```

```
✓ views
  ✓ partials
    ~ heading.hbs
    ~ login-form.hbs
    ~ menu.hbs
    ~ signup-form.hbs
    ~ user-credentials.hbs
    ~ user-details.hbs
    ~ donate.hbs
    ~ layout.hbs
    ~ login.hbs
    ~ main.hbs
    ~ signup.hbs
```

# Key Concepts

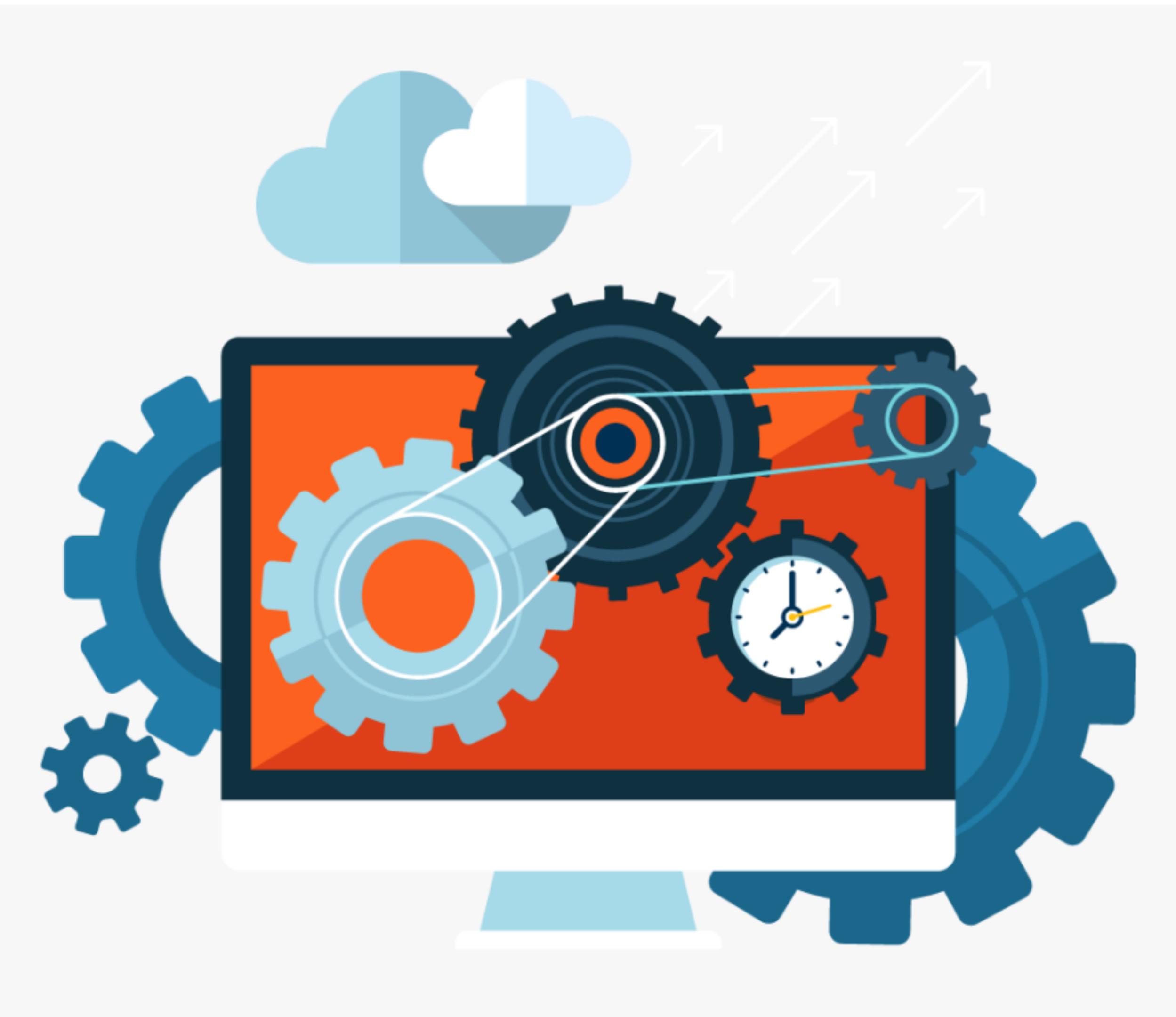
- Routing
- Routing Methods & Parameters
- Templates & Handlebars
- Hapi Plugins
- Views
- Security Strategies - Cookies
- MongoDB
- Mongoose Models





# Application Structure

- Code Quality Tools
- Modules & Scripts
- Server
- Models
- Controllers
- Pages
- Components



# Code Quality Tools

## package.json

```
{  
  "name": "donation-hapi",  
  "version": "0.1.0",  
  "description": "",  
  "main": "src/server.js",  
  "type": "module",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "author": "",  
  "license": "ISC",  
  "devDependencies": {  
    "eslint": "^8.56.0",  
    "eslint-config-airbnb-base": "^15.0.0",  
    "eslint-config-prettier": "^9.1.0",  
    "eslint-plugin-import": "^2.29.1",  
    "nodemon": "^3.0.3",  
    "prettier": "^3.2.4"  
  }  
}
```

Establish application starter including:

- ESLint (AirBnB) linting
- Prettier formatting

## .eslintrc.json

```
{  
  "env": {  
    "node": true,  
    "es2021": true  
  },  
  "extends": ["airbnb-base", "prettier"],  
  "parserOptions": {  
    "ecmaVersion": "latest",  
    "sourceType": "module"  
  },  
  "rules": {  
    "quotes": ["error", "double"],  
    "import/extensions": "off",  
    "import/prefer-default-export": "off",  
    "object-shorthand": "off",  
    "no-unused-vars": "off",  
    "no-underscore-dangle": "off",  
    "no-param-reassign": "off",  
    "no-undef": "off",  
    "import/no-unresolved": "off",  
    "import/no-extraneous-dependencies": "off",  
    "no-console": "off",  
    "func-names": "off"  
  }  
}
```

## .prettierrc.json

```
{  
  "trailingComma": "es5",  
  "tabWidth": 2,  
  "semi": true,  
  "singleQuote": false,  
  "printWidth": 180  
}
```

## package.json

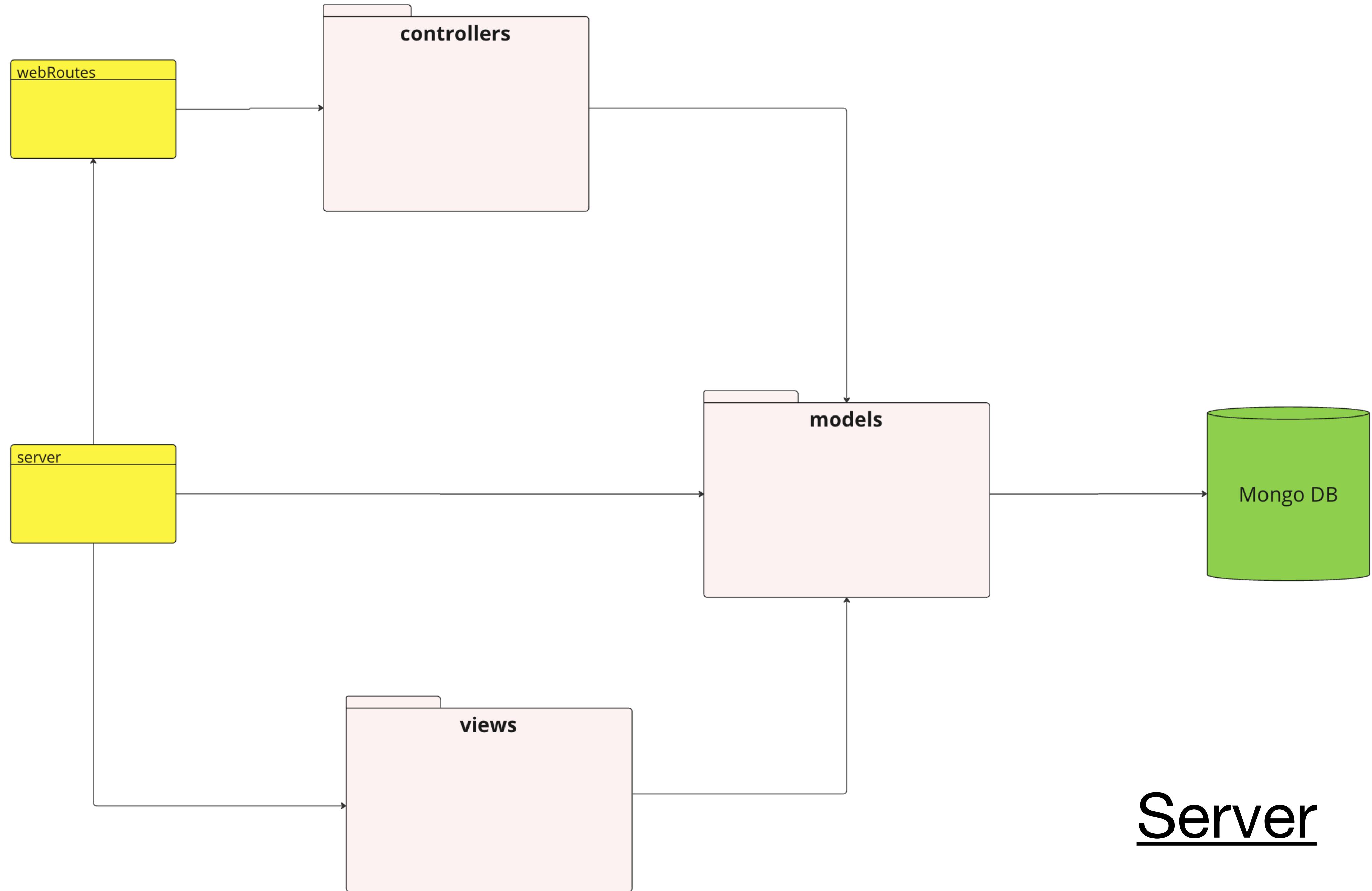
```
"dependencies": {  
    "@hapi/boom": "^10.0.1",  
    "@hapi/cookie": "^12.0.1",  
    "@hapi/hapi": "^21.3.3",  
    "@hapi/inert": "^7.1.0",  
    "@hapi/vision": "^7.0.3",  
    "dotenv": "^16.4.5",  
    "handlebars": "^4.7.8",  
    "mais-mongoose-seeder": "^1.0.7",  
    "mongoose": "^8.2.0"  
}
```

- Hapi Core Libraries + mongoose

## Modules & Scripts

- start, dev, lint & test scripts

```
"scripts": {  
    "start": "node src/server.js ",  
    "dev": "./node_modules/.bin/nodemon src/server.js --ignore 'src/models/json/*.json'",  
    "lint": "./node_modules/.bin/eslint . --ext .js"  
},
```



```

async function init() {
  importEnvs();
  const server = Hapi.server({
    port: process.env.PORT || 4000,
    routes: { cors: true },
  });
  await initPlugins(server);
  initSecurityStrategies(server);
  connectDb("mongo");
  server.route(webRoutes);
  await server.start();
  console.log(`Server running at: ${server.info.uri}`);
}

```

```

import { accountsController } from "./controllers/accounts-controller.js";
import { donationsController } from "./controllers/donations-controller.js";

export const webRoutes = [
  { method: "GET", path: "/", config: accountsController.index },
  { method: "GET", path: "/signup", config: accountsController.showSignup },
  { method: "GET", path: "/login", config: accountsController.showLogin },
  { method: "GET", path: "/logout", config: accountsController.logout },
  { method: "POST", path: "/register", config: accountsController.signup },
  { method: "POST", path: "/authenticate", config: accountsController.login },

  { method: "GET", path: "/donate", config: donationsController.index },

  {
    method: "GET",
    path: "/{param*}",
    handler: {
      directory: {
        path: "./public",
      },
    },
    options: { auth: false },
  },
];

```

- Load .env
- Create server
- Register plugins
- Configure Vision
- Configure cookie auth
- Initialise model (mongo)
- Import routes
- Start

## Server

.env

```

cookie_name=donation-web
cookie_password=secretpasswordnotrevealedtoanyone
db=mongodb://localhost/donation

```

```

function initSecurityStrategies(server) {
  server.auth.strategy("session", "cookie", {
    cookie: {
      name: process.env.cookie_name,
      password: process.env.cookie_password,
      isSecure: false,
    },
    redirectTo: "/",
    validate: accountsController.validate,
  });
  server.auth.default("session");
}

```

- Load .env
  - Configure cookie auth
- Create server
  - Initialise model (mongo)
- Register plugins
  - Import routes
- Configure Vision
  - Start

```

function importEnvs() {
  const result = dotenv.config();
  if (result.error) {
    console.log(result.error.message);
    process.exit(1);
  }
}

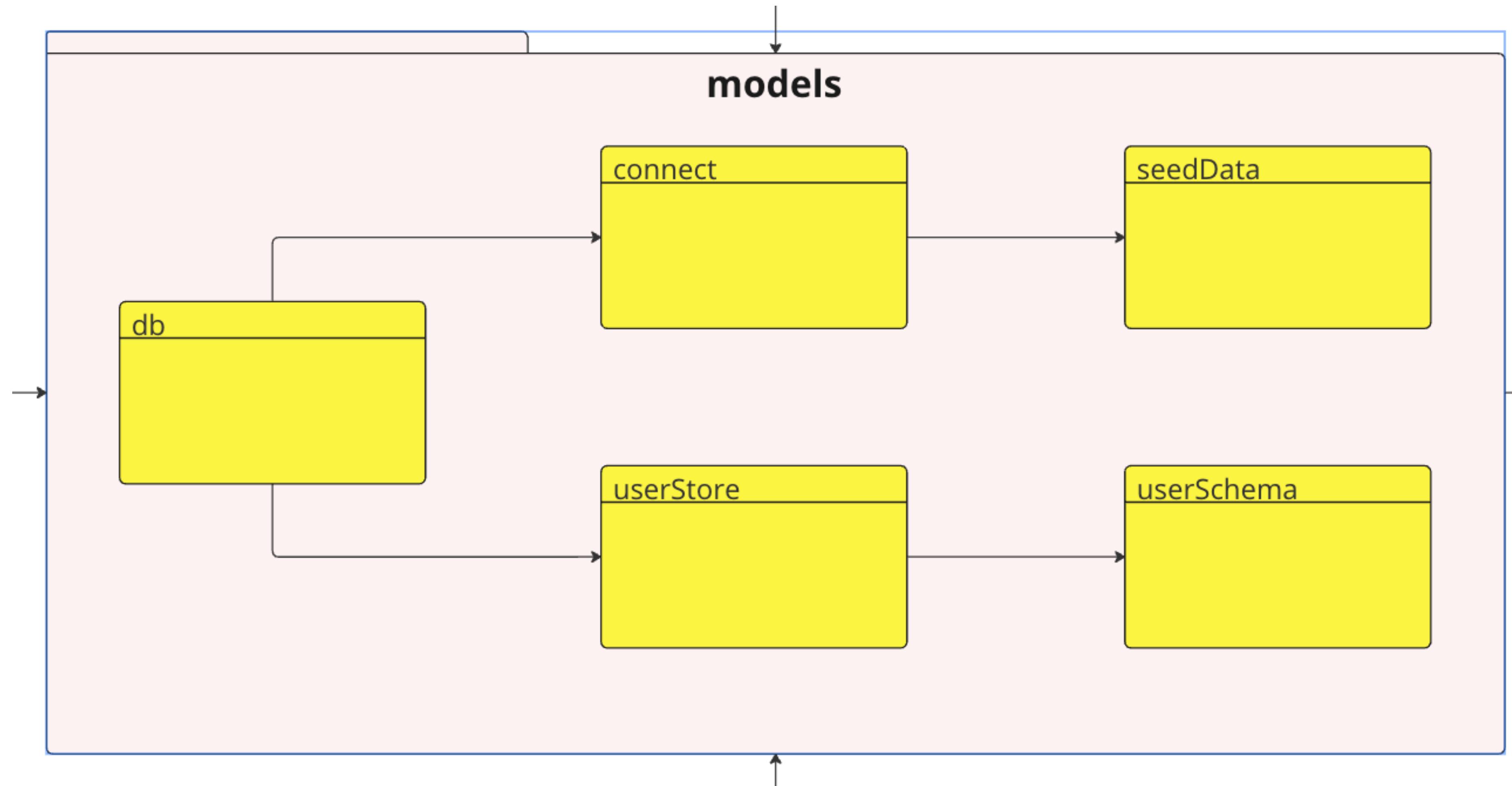
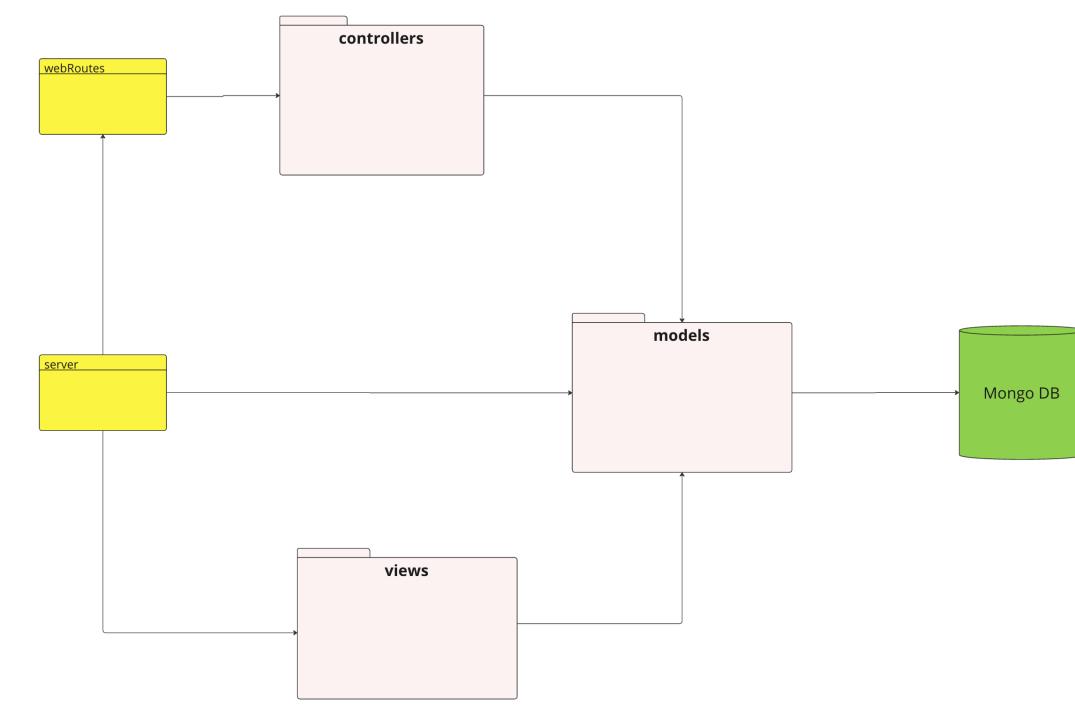
async function initPlugins(server) {
  await server.register(Inert);
  await server.register(Vision);
  await server.register(Cookie);
}

server.views({
  engines: {
    hbs: Handlebars,
  },
  relativeTo: __dirname,
  path: "./views",
  layoutPath: "./views",
  partialsPath: "./views/partials",
  layout: true,
  isCached: false,
});

```

# Server

# Model



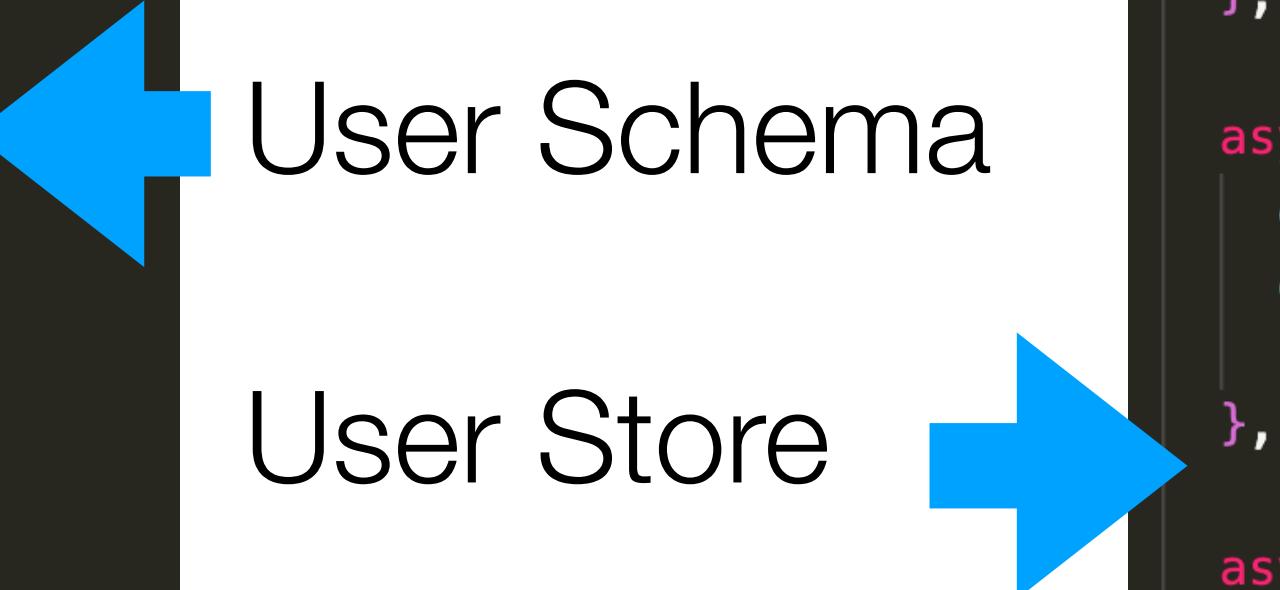
# Models |

```
import Mongoose from "mongoose";

const { Schema } = Mongoose;

const userSchema = new Schema({
  firstName: String,
  lastName: String,
  email: String,
  password: String,
});

export const UserMongoose = Mongoose.model("User", userSchema);
```



```
import { UserMongoose } from "./user.js";

export const userStore = {
  async find() {
    const users = await UserMongoose.find().lean();
    return users;
  },
  async findOne(id) {
    if (id) {
      const user = await UserMongoose.findOne({ _id: id }).lean();
      return user;
    }
    return null;
  },
  async add(user) {
    const newUser = new UserMongoose(user);
    const userObj = await newUser.save();
    return userObj;
  },
  async findBy(email) {
    const user = await UserMongoose.findOne({ email: email }).lean();
    return user;
  },
  async deleteOne(id) {
    try {
      await UserMongoose.deleteOne({ _id: id });
    } catch (error) {
      console.log("bad id");
    }
  },
  async delete() {
    await UserMongoose.deleteMany({});
  },
};
```

```

import { connectMongo } from "./mongo/connect.js";

export const db = {
  userStore: null,
};

export function connectDb(dbType) {
  switch (dbType) {
    case "mongo":
      connectMongo(db);
      break;
    default:
  }
}

```

```

export const seedData = {
  users: [
    {
      _model: "User",
      homer: {
        firstName: "Homer",
        lastName: "Simpson",
        email: "homer@simpson.com",
        password: "secret",
      },
      marge: {
        firstName: "Marge",
        lastName: "Simpson",
        email: "marge@simpson.com",
        password: "secret",
      },
      bart: {
        firstName: "Bart",
        lastName: "Simpson",
        email: "bart@simpson.com",
        password: "secret",
      },
    },
  ],
};

```

- Facade pattern

- Seed data

```

import * as dotenv from "dotenv";
import Mongoose from "mongoose";
import * as mongooseSeeder from "mais-mongoose-seeder";
import { userStore } from "./user-store.js";
import { seedData } from "./seed-data.js";

const seedLib = mongooseSeeder.default;

async function seed() {
  const seeder = seedLib(Mongoose);
  const dbData = await seeder.seed(seedData, { dropDatabase: false, dropCollections: true });
  console.log(dbData);
}

export function connectMongo(db) {
  dotenv.config();

  Mongoose.set("strictQuery", true);
  Mongoose.connect(process.env.db);
  const mongoDb = Mongoose.connection;

  db.userStore = userStore;

  mongoDb.on("error", (err) => {
    console.log(`database connection error: ${err}`);
  });

  mongoDb.on("disconnected", () => {
    console.log("database disconnected");
  });

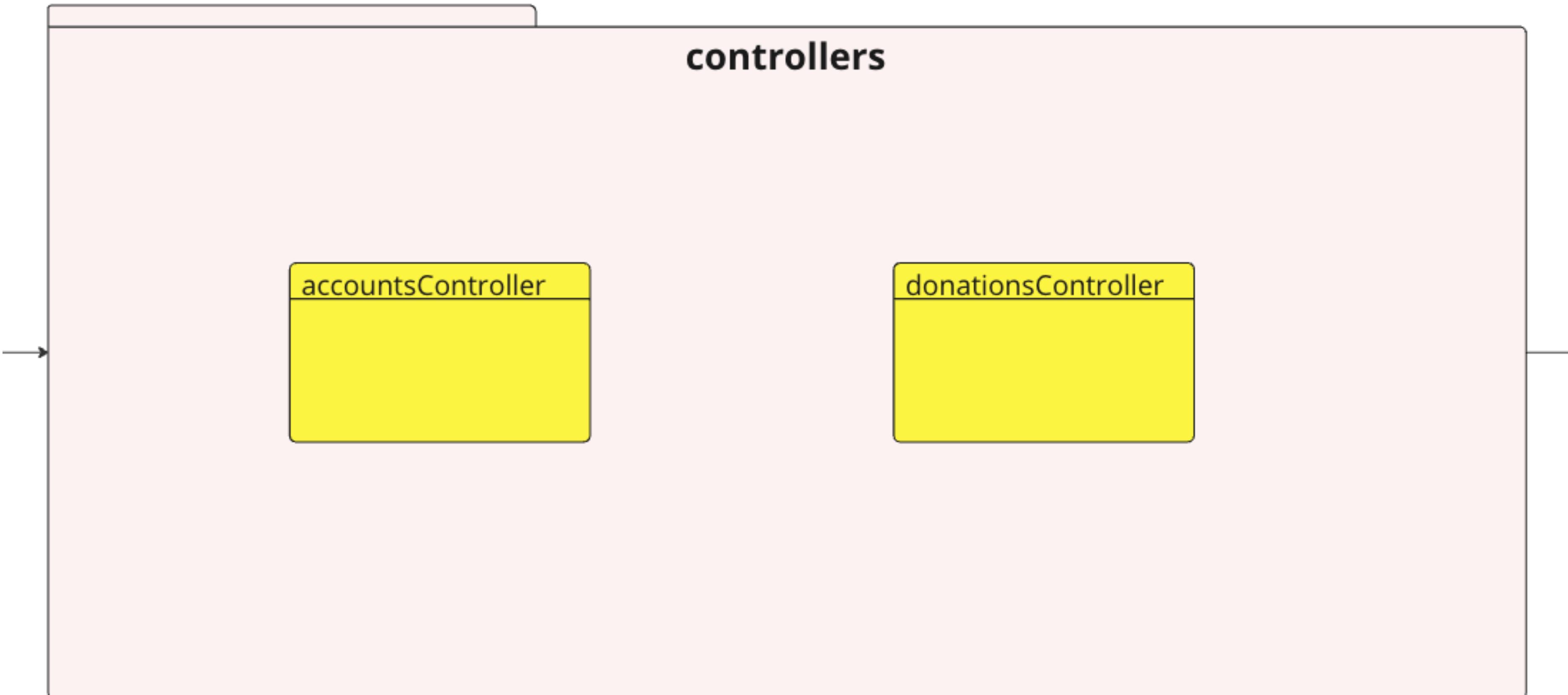
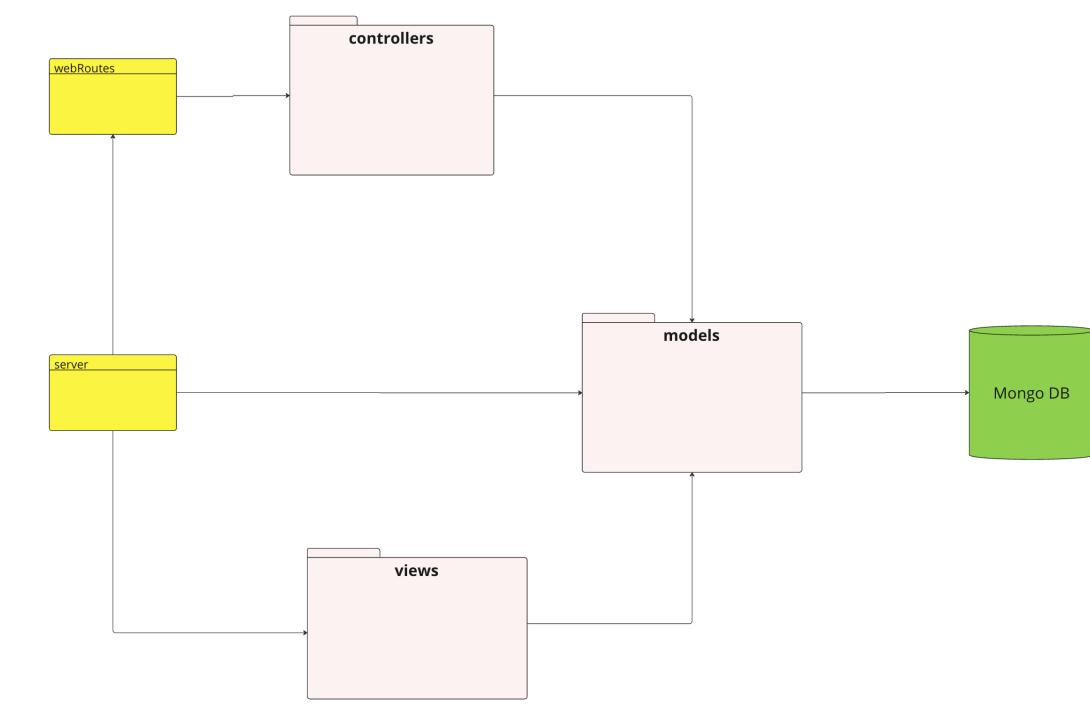
  mongoDb.once("open", () => {
    console.log(`database connected to ${mongoDb.name} on ${mongoDb.host}`);
    seed();
  });
}

```

# Models II

- Initialise database

# Controllers



```
import { db } from "../models/db.js";

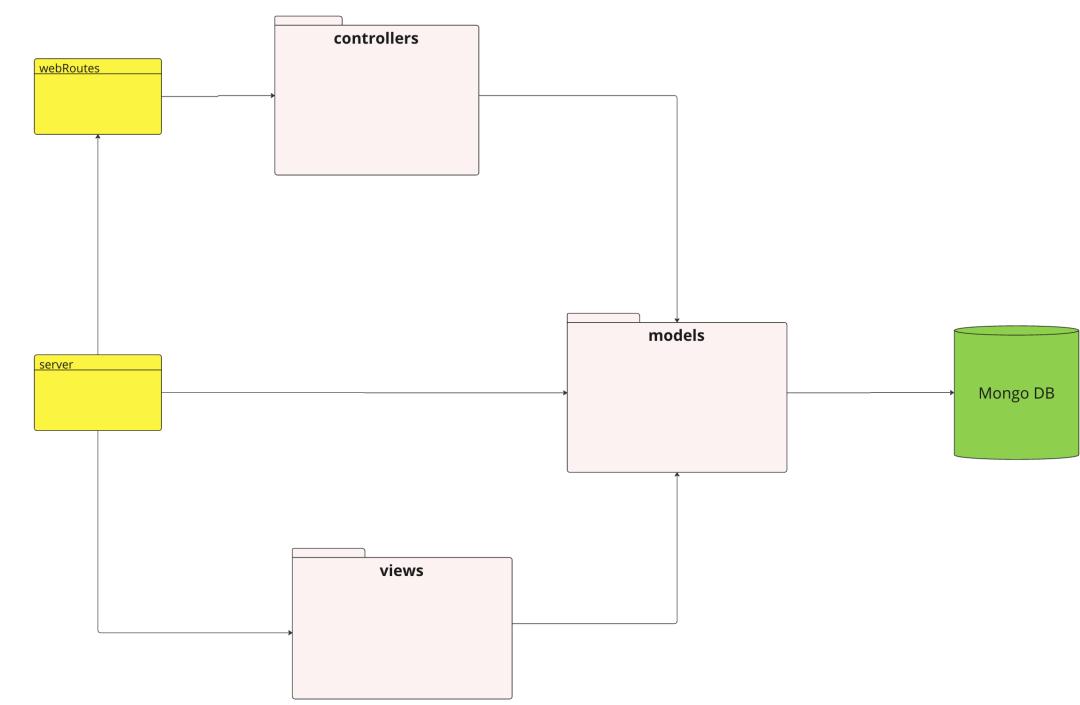
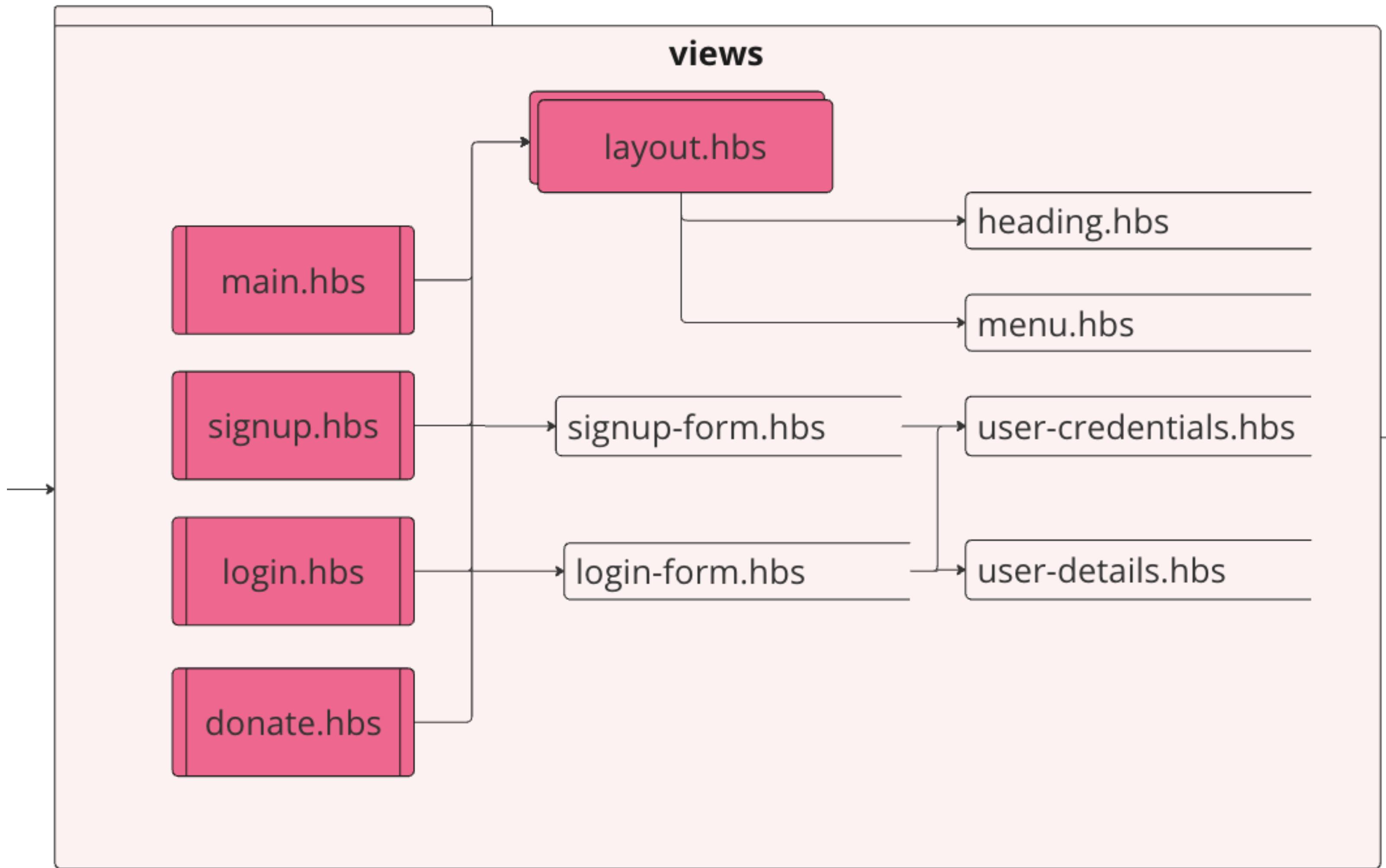
export const accountsController = {
  index: {
    auth: false,
    handler: function (request, h) {
      return h.view("main", { title: "Welcome to Donation" });
    },
  },
  showSignup: {
    auth: false,
    handler: function (request, h) {
      return h.view("signup", { title: "Sign up for Donation" });
    },
  },
  signup: {
    auth: false,
    handler: async function (request, h) {
      const user = request.payload;
      await db.userStore.add(user);
      return h.redirect("/");
    },
  },
  showLogin: {
    auth: false,
    handler: function (request, h) {
      return h.view("login", { title: "Login to Donation" });
    },
  },
  login: {
    auth: false,
    handler: async function (request, h) {
      const { email, password } = request.payload;
      const user = await db.userStore.findBy(email);
      if (!user || user.password !== password) {
        return h.redirect("/");
      }
      request.cookieAuth.set({ id: user._id });
      return h.redirect("/donate");
    },
  },
  logout: {
    handler: function (request, h) {
      request.cookieAuth.clear();
      return h.redirect("/");
    },
  },
  async validate(request, session) {
    const user = await db.userStore.findOne(session.id);
    if (!user) {
      return { isValid: false };
    }
    return { isValid: true, credentials: user };
  },
};
```

# Controllers

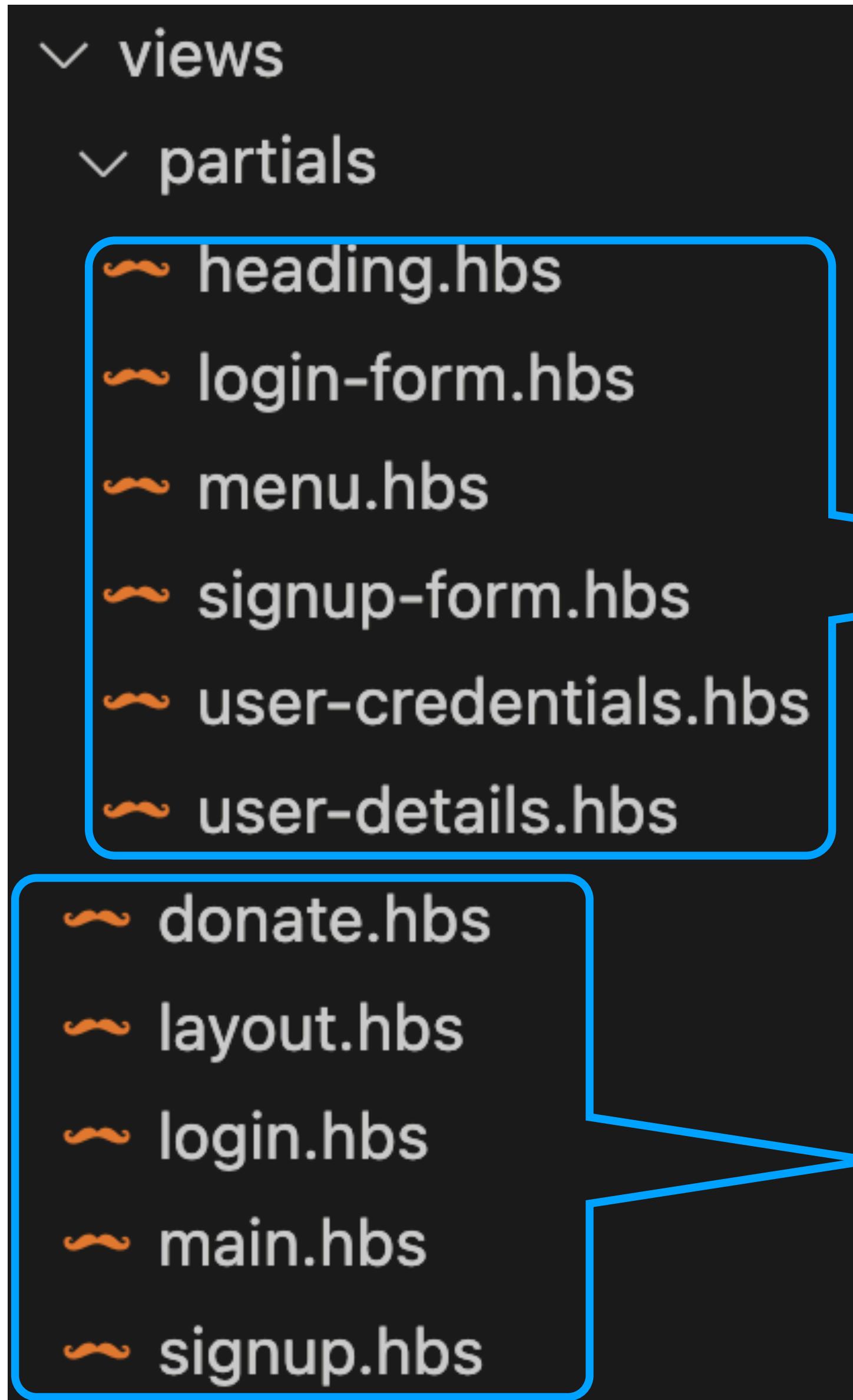
- Standard Users Controller
- Placeholder Donations Controller

```
export const donationsController = {
  index: {
    handler: async function (request, h) {
      const loggedInUser = request.auth.credentials;
      return h.view("donate", { title: "Make a Donation", user: loggedInUser });
    },
  },
};
```

# Views

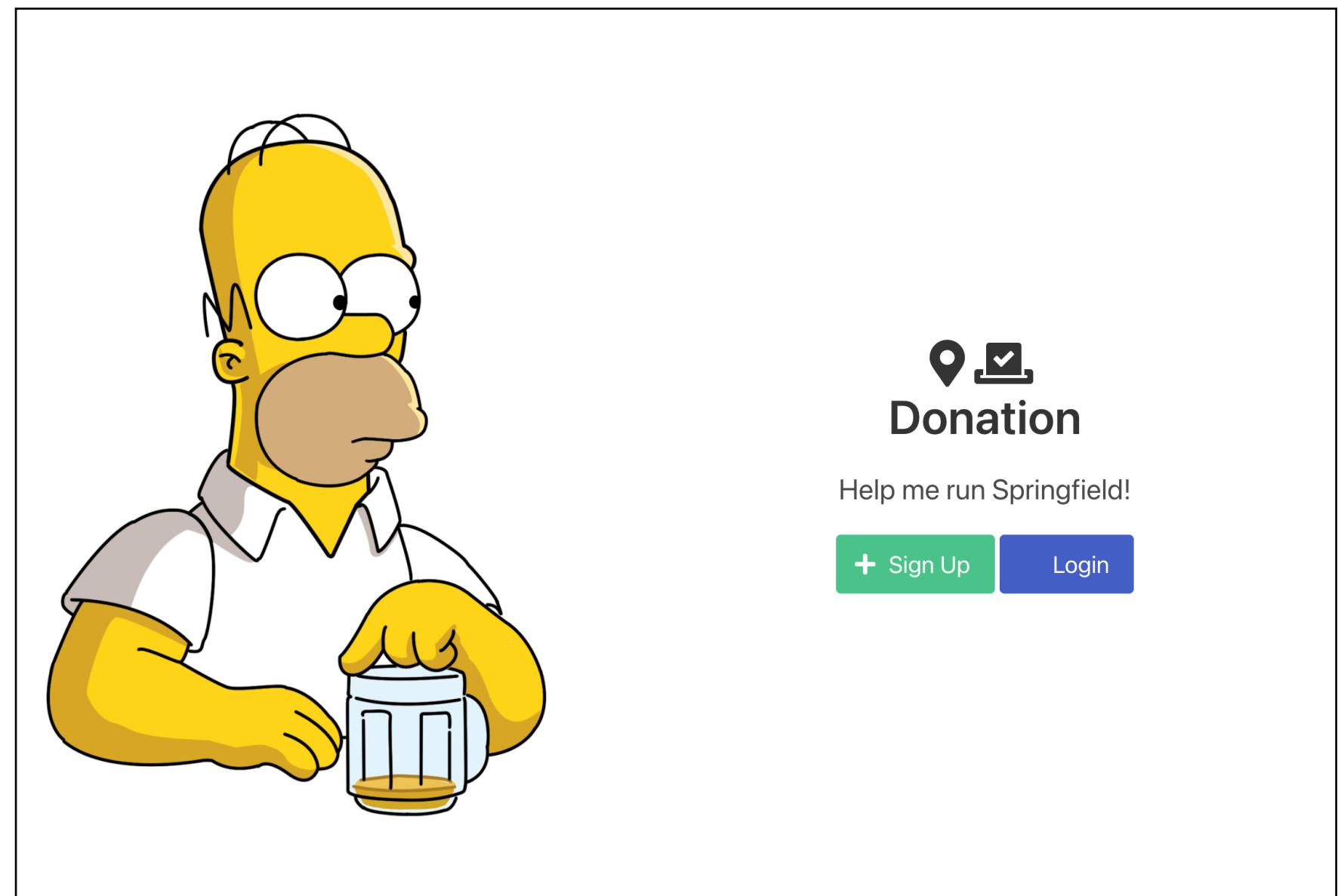


# Views & Partials



# Views

- ~~ donate.hbs
- ~~ layout.hbs
- ~~ login.hbs
- ~~ main.hbs
- ~~ signup.hbs



**Signup to Donation**

**Login to DONATION**

**Email**

**Password**

**Log In**

**Name**

 First Name  Last Name

**Email**

**Password**

**Create Account**

Already have an account? [Login Here](#)

# layout.hbs

```
<!DOCTYPE html>
<html>

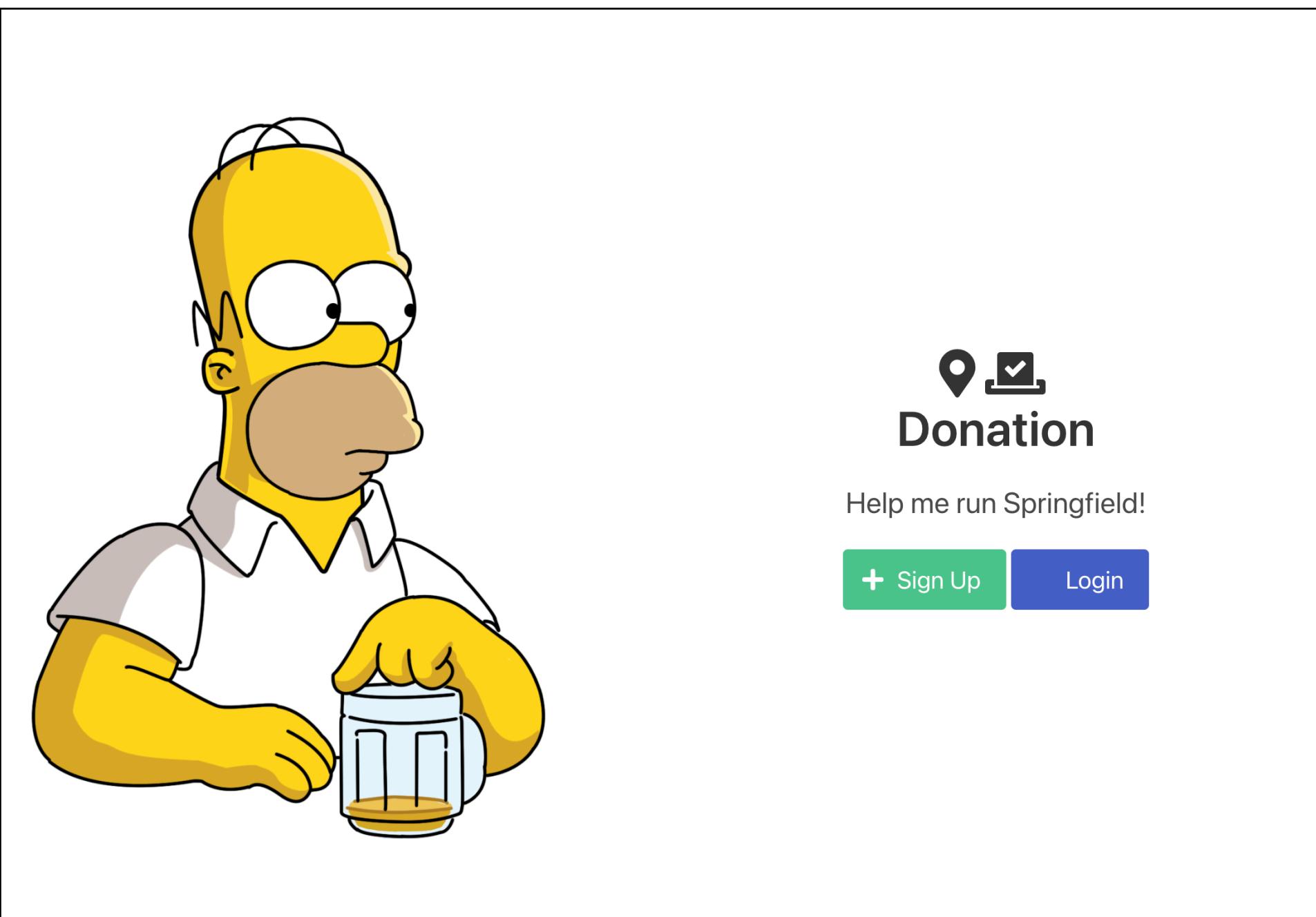
<head>
  <title>{{title}}</title>
  <meta charset="UTF-8" />
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.4/css/bulma.min.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/css/all.min.css"
    integrity="sha512-+4zCK9k+qNFUR5X+cKL9EIR+Z0htIloNl9GIKS57V1MyNsYpYcUrUeQc9vNfzsWfV28IaLL3i96P9sdNyeRssA=="
    crossorigin="anonymous" />
  <link href="https://fonts.googleapis.com/css?family=Open+Sans:400,600,300" rel="stylesheet" type="text/css" />
</head>

<body>
  <div class="container">
    {{content}}
  </div>
</body>

</html>
```

# splashscreen.hbs

```
<section class="hero is-fullheight is-default is-bold">
  <div class="hero-head"></div>
  <div class="hero-body">
    <div class="container has-text-centered">
      <div class="columns is-vcentered">
        <div class="column is-5">
          <figure class="image">
            
          </figure>
        </div>
        <div class="column is-6 is-offset-1">
          <h1 class="title is-2 mb-5">
            <span class="icon mr-5"><i class="fas fa-map-marker-alt"></i></span><span class="icon mr-4">
              <i class="fas fa-vote-yea"></i></span>
            <p>Donation</p>
          </h1>
          <h2 class="subtitle is-4 mt-2">Help me run Springfield!</h2>
          <p class="has-text-centered">
            <a class="button is-medium is-success" href="/signup">
              <span class="icon"><i class="fas fa-plus"></i></span><span>Sign Up</span>
            </a>
            <a class="button is-medium is-link" href="/login">
              <span class="icon"><i class="fas fa-sign-in"></i></span><span>Login</span>
            </a>
          </p>
        </div>
      </div>
    </div>
  </div>
</section>
```



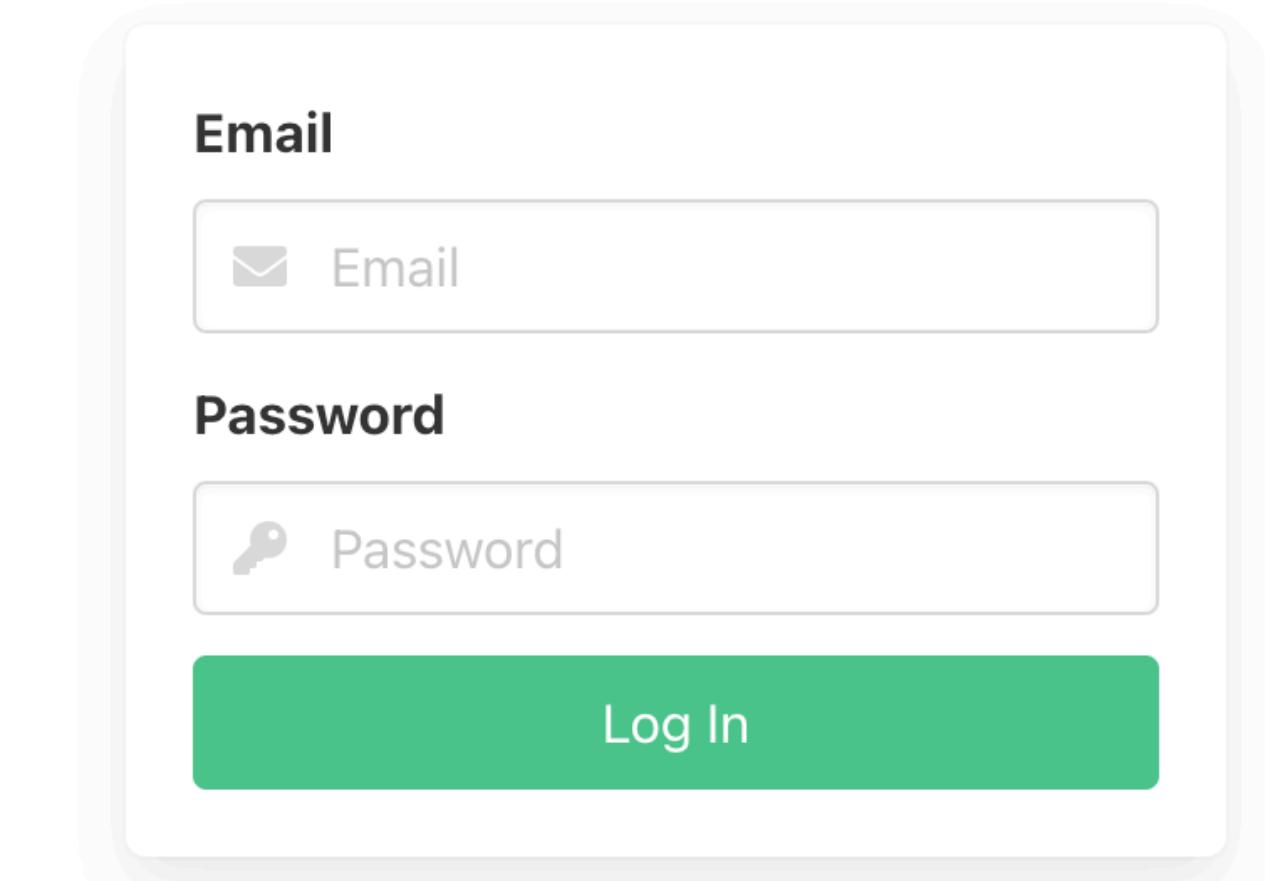
# main.hbs

```
{ {> splashscreen } }
```

## user-credentials.hbs

```
<div class="field">
  <label class="label">Email</label>
  <div class="control has-icons-left">
    <input class="input" id="email" type="text" placeholder="Enter email" name="email">
    <span class="icon is-small is-left">
      <i class="fa fa-envelope"></i>
    </span>
  </div>
</div>
<div class="field">
  <label class="label">Password</label>
  <div class="control has-icons-left">
    <input class="input" id="password" type="password" placeholder="Enter Password" name="password">
    <span class="icon is-small is-left">
      <i class="fa fa-key"></i>
    </span>
  </div>
</div>
```

## Login to DONATION



The form is a light gray rounded rectangle with a thin shadow. It features two input fields: one for 'Email' with a small envelope icon and one for 'Password' with a small key icon. A large green 'Log In' button is at the bottom.

Email

Password

Log In

## login-form.hbs

```
<form action="/authenticate" method="POST">
  {{> user-credentials }}
  <button class="button is-success is-fullwidth">Log In</button>
</form>
```

## login.hbs

```
<section class="hero is-fullheight">
  <div class="hero-body">
    <div class="container">
      <div class="column is-4 is-offset-4">
        <h3 class="title has-text-black has-text-centered">Login to DONATION</h3>
        <div class="box">
          {{> login-form }}
        </div>
      </div>
    </div>
  </div>
</section>
```

## user-details.hbs

```
<label class="label">Name</label>
<div class="field is-horizontal">
  <div class="field-body">
    <div class="field">
      <div class="control has-icons-left">
        <input class="input" id="firstname" type="text" placeholder="Enter first name" name="firstName">
        <span class="icon is-small is-left">
          <i class="fa fa-id-card"></i>
        </span>
      </div>
    </div>
    <div class="field">
      <div class="control has-icons-left">
        <input class="input" id="lastname" type="text" placeholder="Enter last name" name="lastName">
        <span class="icon is-small is-left">
          <i class="fa fa-id-card"></i>
        </span>
      </div>
    </div>
  </div>
</div>
```

## Signup to Donation

Name

First Name

Last Name

Email

Email

Password

Password

Create Account

Already have an account? [Login Here](#)

## signup-form.hbs

```
<form action="/register" method="POST">
  {{> user-details }}
  {{> user-credentials }}
  <button class="button is-success is-fullwidth">Create Account</button>
  <br />
</form>
```

## signup.hbs

```
<section class="hero is-fullheight">
  <div class="hero-body">
    <div class="container">
      <div class="column is-4 is-offset-4">
        <h3 class="title has-text-black has-text-centered">Login to DONATION</h3>
        <div class="box">
          {{> signup-form }}
        </div>
      </div>
    </div>
  </div>
</section>
```

## menu.hbs

```
<nav class="navbar is-full-width">
  <div class="container">
    <div class="navbar-brand">
      <a class="navbar-item" href="/dashboard">
        <span class="icon"> <i class="fas fa-map-marker-alt"></i></span><span class="icon mr-1">
          <i class="far fa-money-bill-alt"></i></span><span><strong>Donation</strong></span>
        </a>
    </div>
    <div id="navbarMenu" class="navbar-menu">
      <div class="navbar-end">
        <a class="navbar-item" href="/logout"> Logout <span class="is-size-7">({{user.email}})</span></a>
      </div>
    </div>
  </div>
</nav>
```

## heading.hbs

```
<section class="hero is-link is-small mt-4 mb-4">
  <div class="hero-body">
    <div class="container">
      <h1 class="title has-text-centered is-size-4">
        {{title}}
      </h1>
    </div>
  </div>
</section>
```

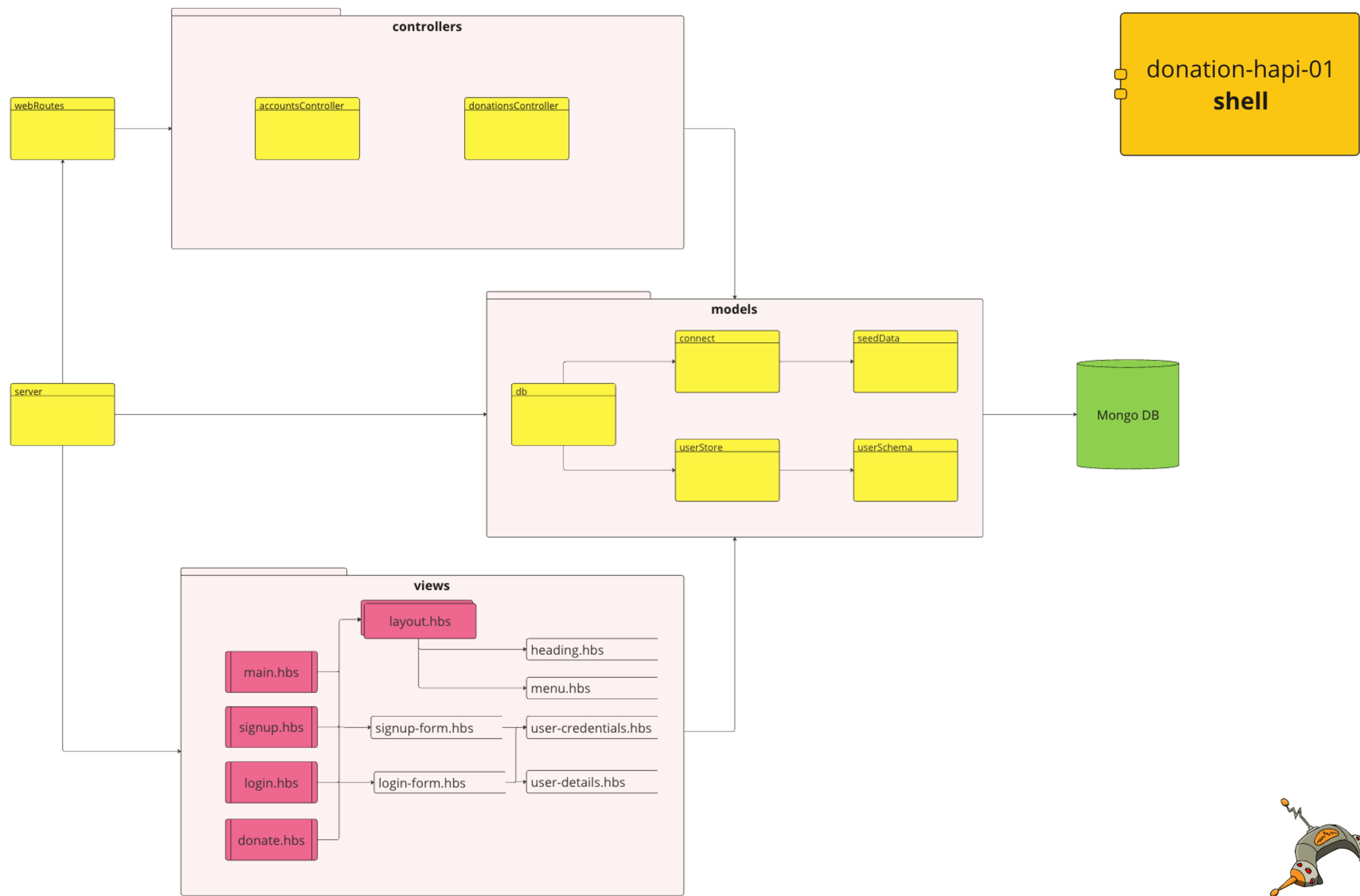
```
{{> menu }}
{{> heading}}
```

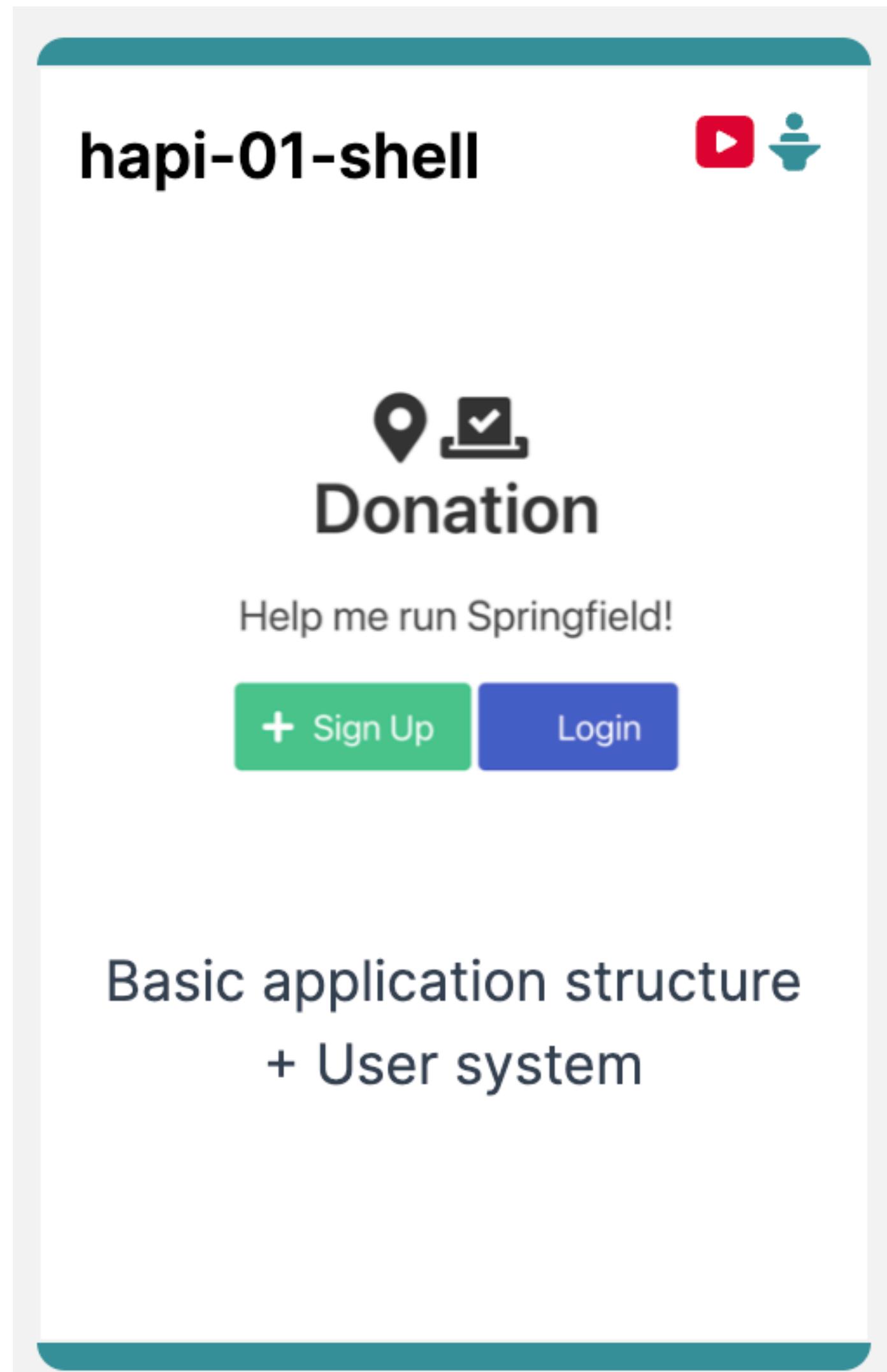
## donate.hbs

📍 💸 Donation

Logout(homer@simpson.com)

Make a Donation





Basic application structure

+ User system