[REPORT]

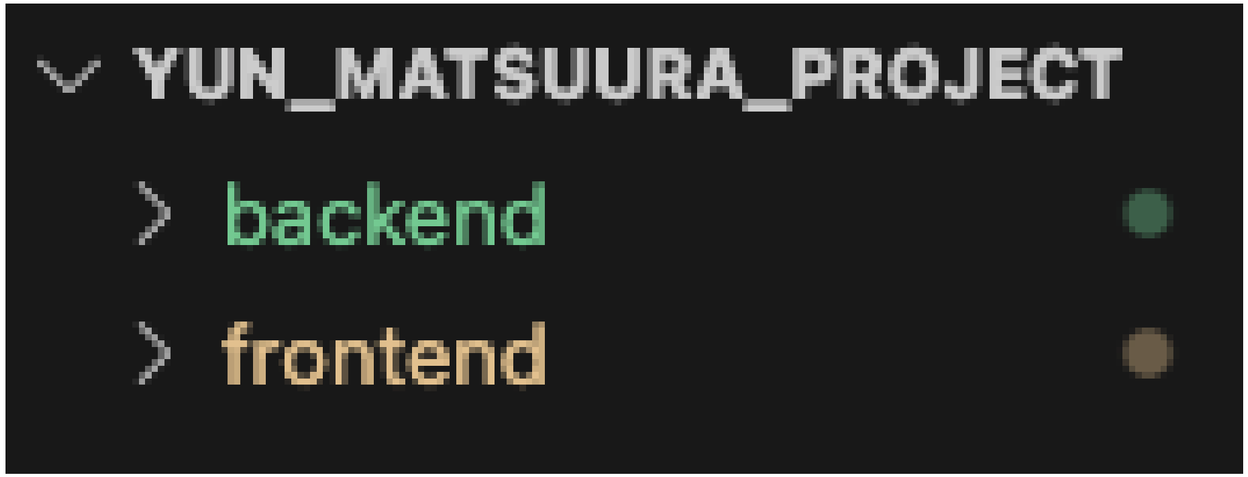
Full Stack Project

Yun Matsuura

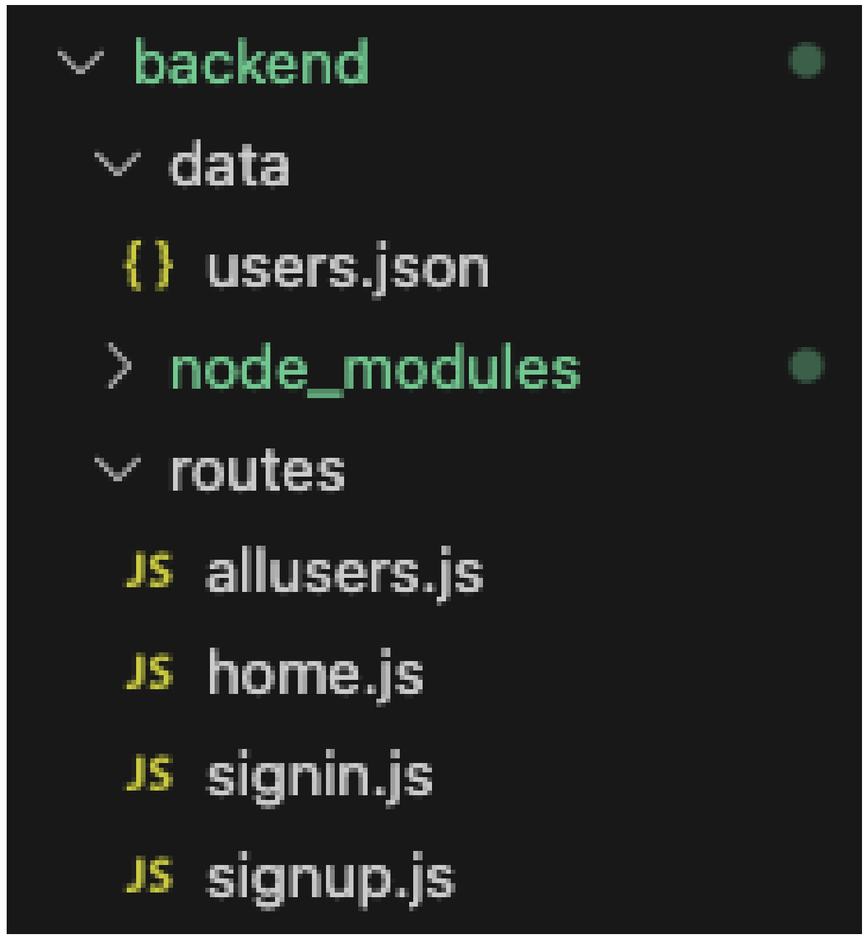
Jul. 30th, 2025

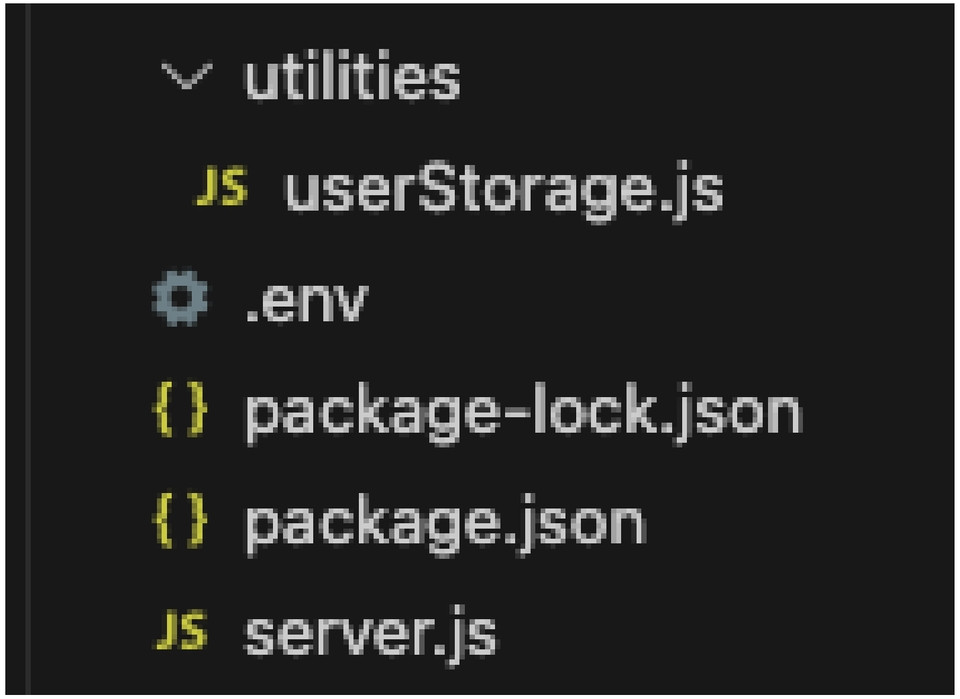
[Index]

[React Application Architecture]

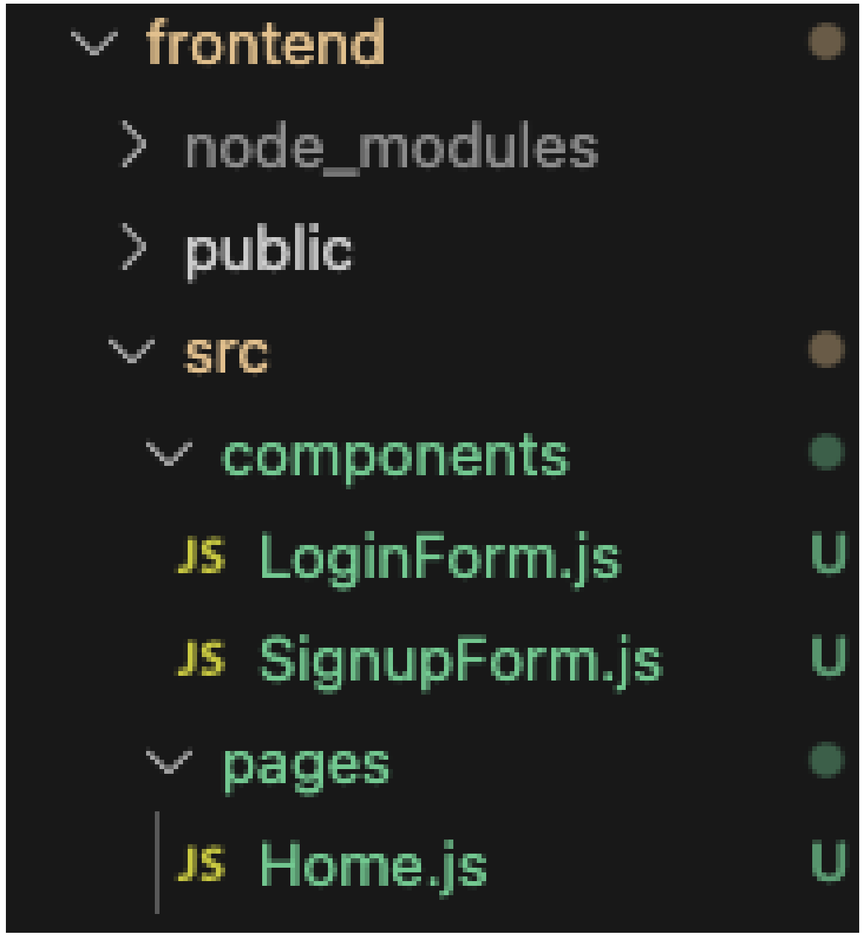


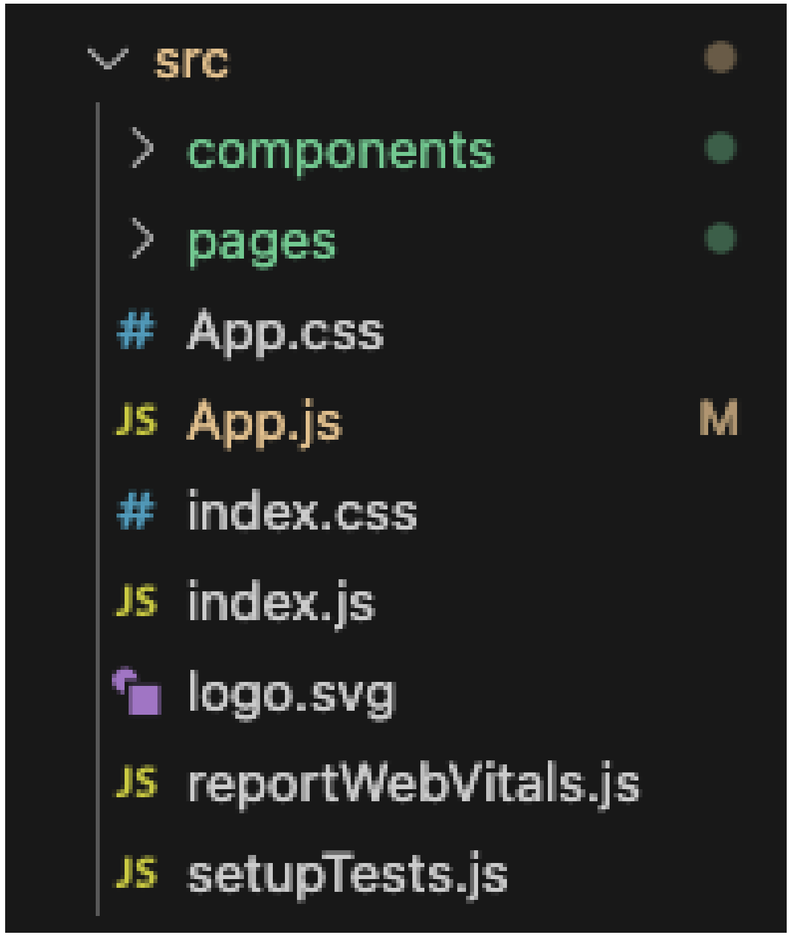
[React Application Architecture]

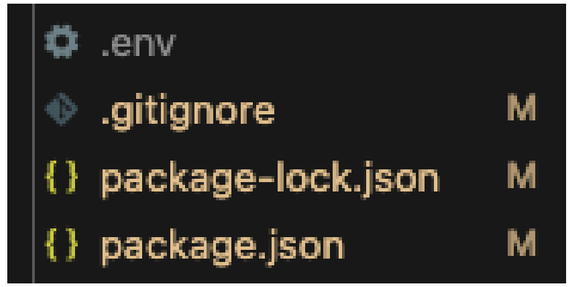




[React Application Architecture]







[Frontend Routing]

```
function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<LoginForm />} />
        <Route path="/signup" element={<SignupForm />} />
        <Route path="/home" element={<Home />}/>
      </Routes>
    </BrowserRouter>
  );
}
```

App.js has Routing
No nested or dynamic routing

[Frontend Routing]

useState :

to manage data of the login user on sign up, sign in

```
function LoginForm() {
  const [inputEmail, setInputEmail] = useState("");
  const [inputPassword, setInputPassword] = useState("");
```

```
<input
  id="emailInput"
  type="email"
  required
  value={inputEmail}
  onChange={(e) => setInputEmail(e.target.value)}
/>
```

[Frontend Routing]

useNavigate : signIn/signUp => home

```
const navigate = useNavigate();
```

```
navigate("/home", { state: { user: data.user}});
```

useLocation :
to receive user info from LoginForm/SignupForm

```
function Home() {
  const location = useLocation();
  const [users, setUsers] = useState([]);
  const [showUsers, setShowUsers] = useState(false);

  const user = location.state?.user;
```

```
navigate("/home", { state: { user: data.user}});
```

[Frontend Routing]

useState :

setUsers : fetched data of all users

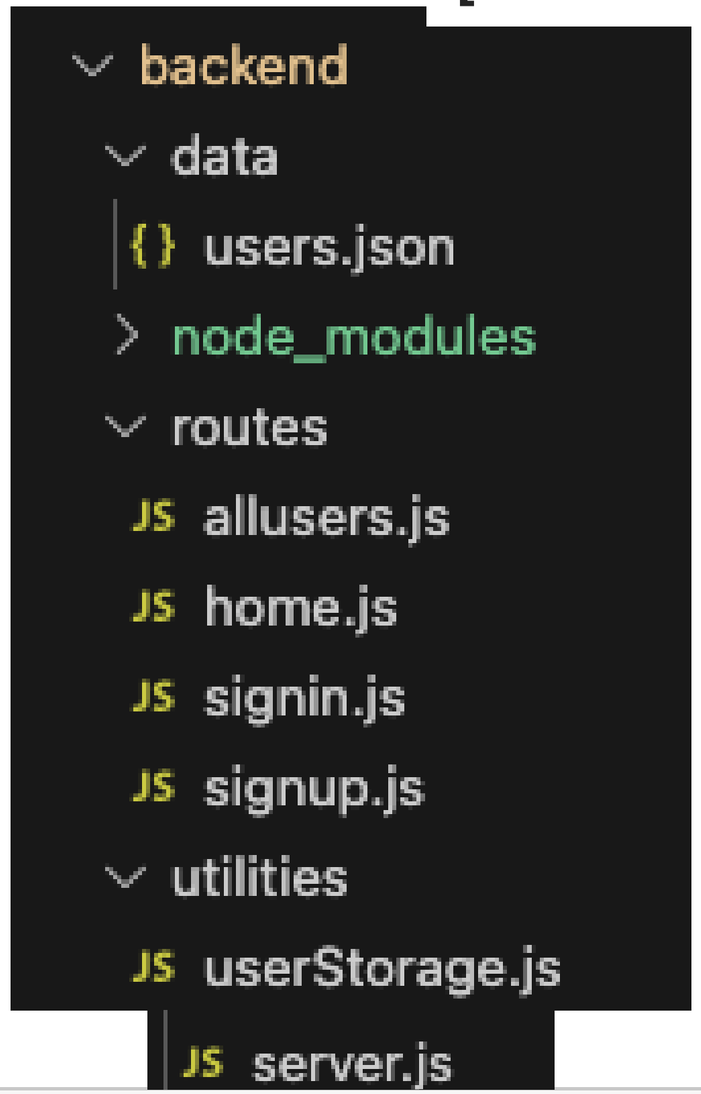
setShowUsers : true/false to show all users

```
function Home() {
  const location = useLocation();
  const [users, setUsers] = useState([]);
  const [showUsers, setShowUsers] = us

  const user = location.state?.user;
```

```
try {
  const response = await fetch(
    `${process.env.REACT_APP_URL}/home/all-users`
  );
  const data = await response.json();
  setUsers(data);
  setShowUsers(true);
} catch (e) {
  console.error("Falied to fetch users");
}
```

[Node-Express Server Overview]



data : all user information

routes : each file has routing action

utilities : common functionalities

server.js : all routing

```
app.use(express.json());
app.get("/allusers", allUsers);
app.use("/signup", signUp);
app.use("/signin", signIn);
app.use("/home", home);
```

[Node-Express Server Overview]

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

```
router.post(
  "/",
  signUpValidations,
  validationResponder,
  checkEmailExistance,
  hashPassword,
  (req, res) => {
    const { firstName, l
```

example of signup

server.js

```
app.get("/allusers", allUsers);
app.use("/signup", signUp);
app.use("/signin", signIn);
app.use("/home", home);
```

```
router.get("/all-users", (req, res) => {
  const usersPath = path.join(__dirname, "../data/users.json");
```

example of home

```
//importing Routes
const allUsers = require("./routes/allusers");
const signUp = require("./routes/signup");
const signIn = require("./routes/signin");
const home = require("./routes/home");
```

[Middleware Usage]

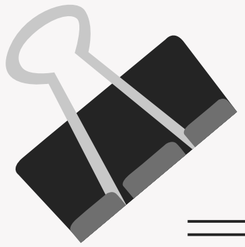
check email exists on signin

```
// Middleware to check if the email exists
const checkEmailExistance = (req, res, next) => {
  const submittedEmail = req.body.email;

  const existingUsers = getAllUsers();

  const userExist = existingUsers.find((user) => user.email === submittedEmail);

  if (!userExist) {
    const error = new Error("Email not found. Please try again or sign up.");
    error.status = 400;
    next(error);
  } else {
    req.user = userExist;
    next();
  }
};
```



[Middleware Usage]

check email exists on signin

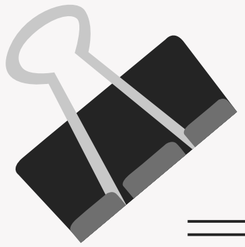
```
// Middleware to check if the email exists
const checkEmailExistance = (req, res, next) => {
  const submittedEmail = req.body.email;

  const existingUsers = getAllUsers();

  const userExist = existingUsers.find((user) => user.email === submittedEmail);

  if (!userExist) {
    const error = new Error("Email not found. Please try again or sign up.");
    error.status = 400;
    next(error);
  } else {
    req.user = userExist;
    next();
  }
};
```

same name middleware on signup but slightly different action



[Middleware Usage]

signup

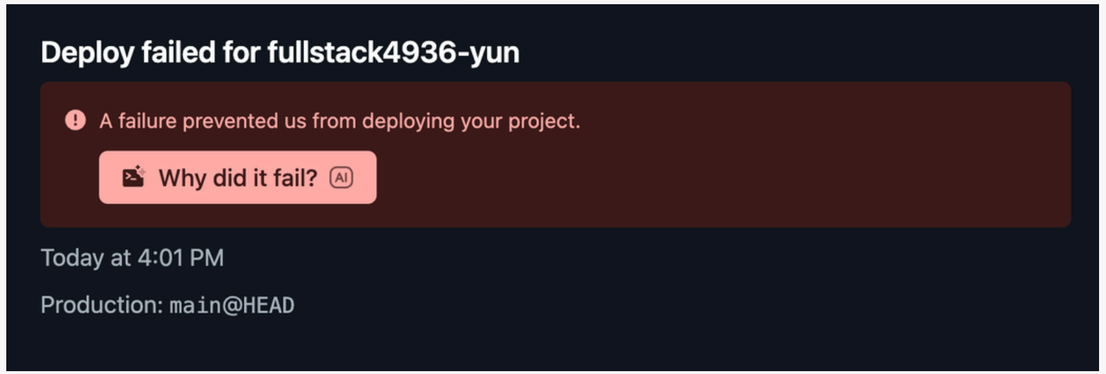
```
router.post(
  "/",
  signUpValidations,
  validationResponder,
  checkEmailExistance,
  hashPassword,
  (req, res) => {
    const { firstName, lastNam
```

```
const signUpValidations = [
  body("firstName").notEmpty().withMessage("First name is required.")
  body("email").isEmail().withMessage("Invalid email"),
  body("password")
    .isLength({ min: 5 })
    .withMessage("Pass word should be at least 5 characters."),
  body("repassword")
    .custom((value, { req }) => value === req.body.password)
    .withMessage("Passwords do not match"),
];
```

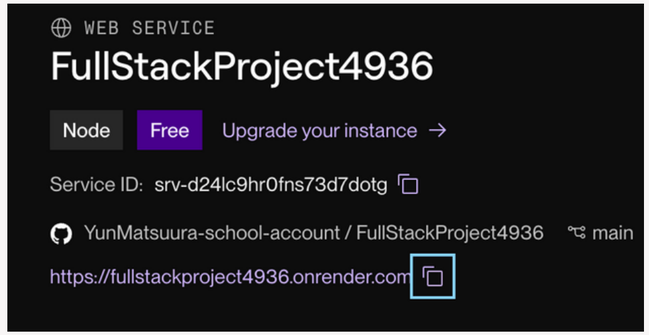
signUpValidation is array for validations

[Middleware Usage]

Frontend : Netlify



Backend : Render



Environment Variables :
stored in .env.
.end is in .gitingore
set variables manually in Render