

**Supervisor**: Junaid Nasir

**Employer**: Abdul Baseer Qazi

**Internee**: Humza Asrar A

**Project Title:** EcoPanel Marketplace

**Project Description:**

A comprehensive React-based web application designed to facilitate the **buying, selling, and management of solar panels** (and renewable energy solutions [achieved later]). This platform serves as a centralized hub for various stakeholders in the solar panel industry {Suppliers(sellers), Customers(buyers), Admin(agent/facilitator), Teams(internal-electrification, outdoor-fixers)}.

**Technologies Used:**

1. Frontend Framework:
   1. React.js
2. UI Libraries:
   1. Material-UI (Button component)
3. State Management:
   1. React hooks (useState)
4. Styling:
   1. CSS-in-JS solutions like styled-components or Emotion

(Sass or Less can be used for advanced styling; [NOT USED])

1. Testing Frameworks:
   1. Jest (JavaScript pre-configured default testing framework)
2. File Input Handling:
   1. HTML input elements
3. Form Submission:

a. FormData API

1. Image Processing:

a. Cloudinary (for image uploads of solar panels by *supplier or internal teams*)

1. Build Tools:

a. Webpack (module bundler)

Can be used but NOT USED: [Babel (JavaScript compiler)]

1. Version Control:
   1. Git
2. Package Manager:
   1. NPM
3. Build Process:
   1. Create React App (for project setup and configuration)
4. API Integration:
   1. Fetch API (native JavaScript fetch functionality)
5. Database:
   1. MySQL (for querying and relationships)
6. Backend Framework:
   1. Express.js (Node.js framework for building RESTful APIs)
7. Authentication:
   1. BcryptJs for hash-based authentication.

Also possible to use but [NOT USED]:

[JWT (JSON Web Tokens) for secure token-based authentication]

[Passport.js (authentication middleware for Node.js)]

1. Deployment & CI/CD :
   1. Local uptill now;

(Docker, Kubernetes, GH CodeSpace can be used)

1. Routing:

a. React Router v6 (BrowserRouter, Routes, Route)

1. CORS Middleware:

a. cors

1. Database Connection:

a. mysql2/promise

1. Email Service:

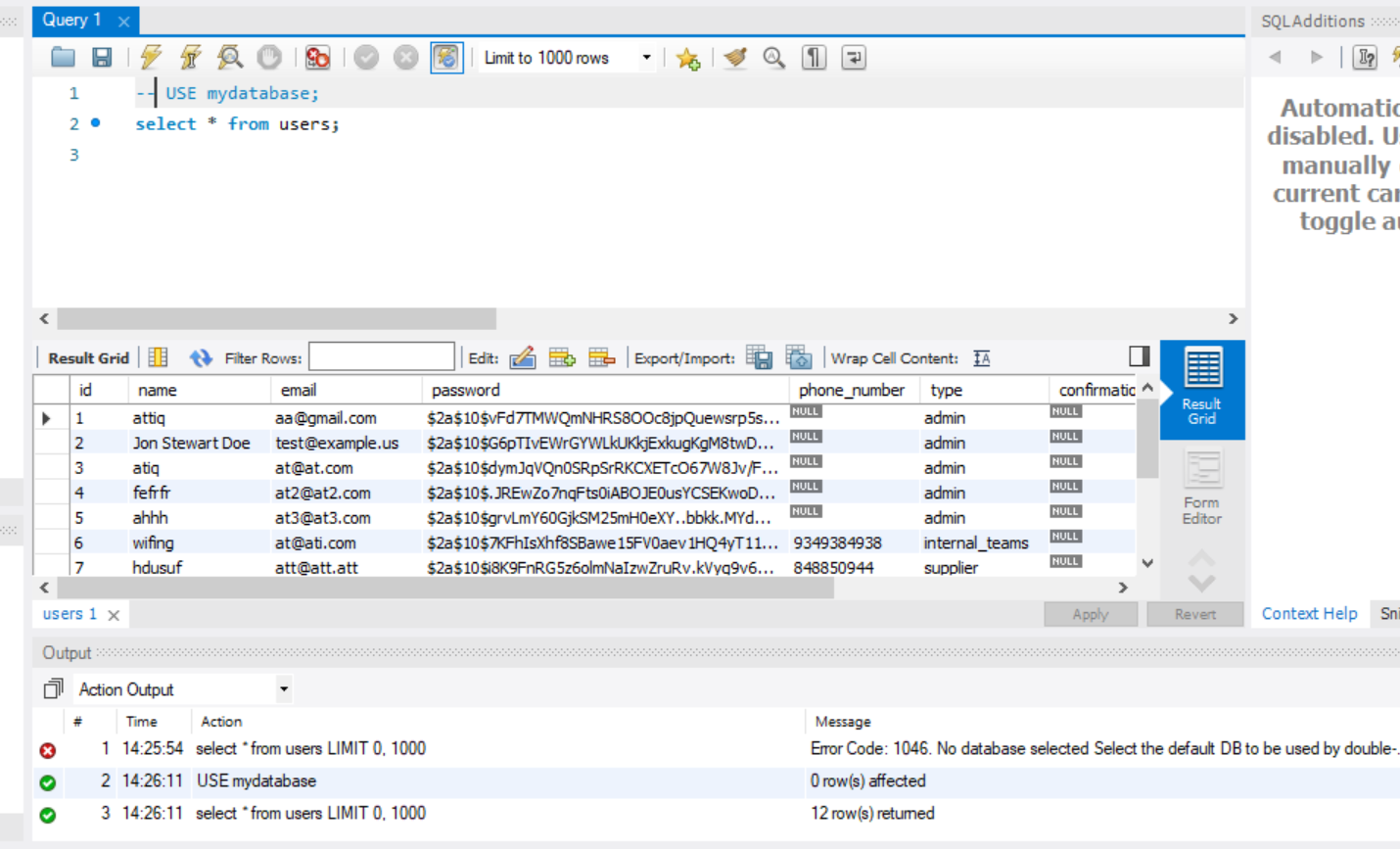
a. Nodemailer (for email verification [primary code added but functionality not utilized])

**Features:**

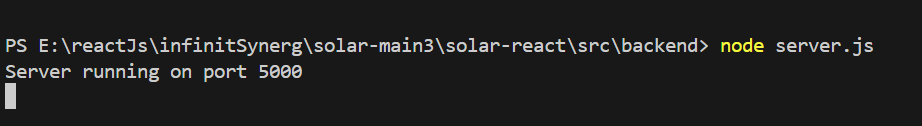
1. [ADDED] Signup screen with different dropdown option for admin, supplier, customer, internal\_Team, as ‘type’ stored in DB that based upon selection.
2. [NOT ADDED] During Sign-in, the authenticator would check what the ‘type’ is in DB, hence based upon that, traverse to the appropriate screen having some additional functionalities/features in it based upon the type.
3. [NOT ADDED] Signup for admin, internal\_teams, supplier would require an additional verification step through using ‘nodemailer’ that would require verification and ‘request accepting’ by the main (hardcoded) administrator.
4. [ADDED] Database to store and handle credentials and details.
5. [NOT ADDED] Database(or another table in the same DB) to store products with their specifications.
6. [ADDED] Homescreen, Contactpage, Aboutpage APIs and webpage upon click
7. [NOT ADDED] Corrected beautified frontend along with correct display of products ad features as per requirements.
8. [NOT ADDED] Customer can traverse from the product clicking ‘buy now’ to loginPage (if not login), else direct to cart/order frame.
9. [88% NOT ADDED] ‘Supplier’ after login can post description+pics (achieved through **Cloudinary** for storing globally) of panels that will be on pending status until admin doesn’t approve it. When ‘admin’ approves it, it would resultingly be showed for ‘customers’ as well as in ‘admins’ project directory. *{only separate SupplierPage made with upload description+pics frontend only (nonfunctional)}*
10. [NOT ADDED] Personal details can be changed or added in ‘settings’ along with password change.
11. [NOT ADDED] Payment, order history and net-metering details, profit calculation can be viewed after logging in by ‘customer’ and ‘supplier’ of his own, and all by admin account-wise.
12. [NOT ADDED] ‘Internal teams’ will have a checklist passed by the admin. When they check 1 item, camera/gallery opens and they upload/capture the pic to tick it the item/work, in the list. Admin will receive notification of that ticked item with photo and an option to add more custom stuff to the list.
13. [NOT ADDED] ‘Tables’ in DB for 3 previously mentioned stuff.
14. [NOT ADDED] A web-scraper/crawler component that scraps certain data from specific websites or excel sheet, and the data is then changeable/editable by the admin. Some % would be added to the price/cost for profit then displayed to the customer.

**Github Link:** [**https://github.com/HumzAhme/solar-react**](https://github.com/HumzAhme/solar-react)

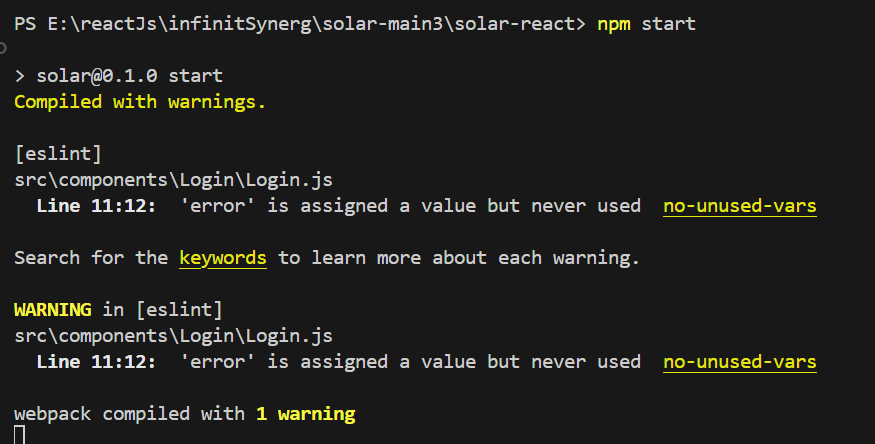
**Mockups Screenshots:**



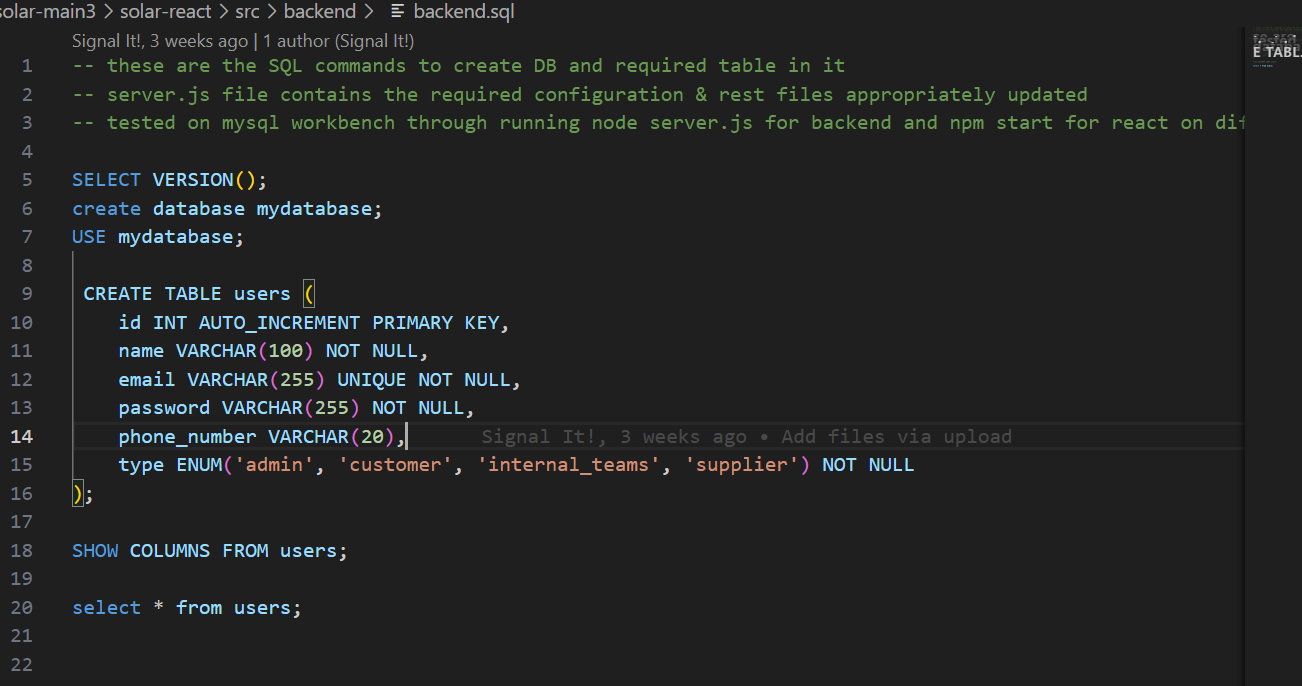
Database in MySQL, can be created by running the commands in ‘*backend.sql*’ (the current data is local to me only populated for feature testing only)



Run ‘node server.js’ in ‘backend’ folder to start backend after starting the DB in workbench

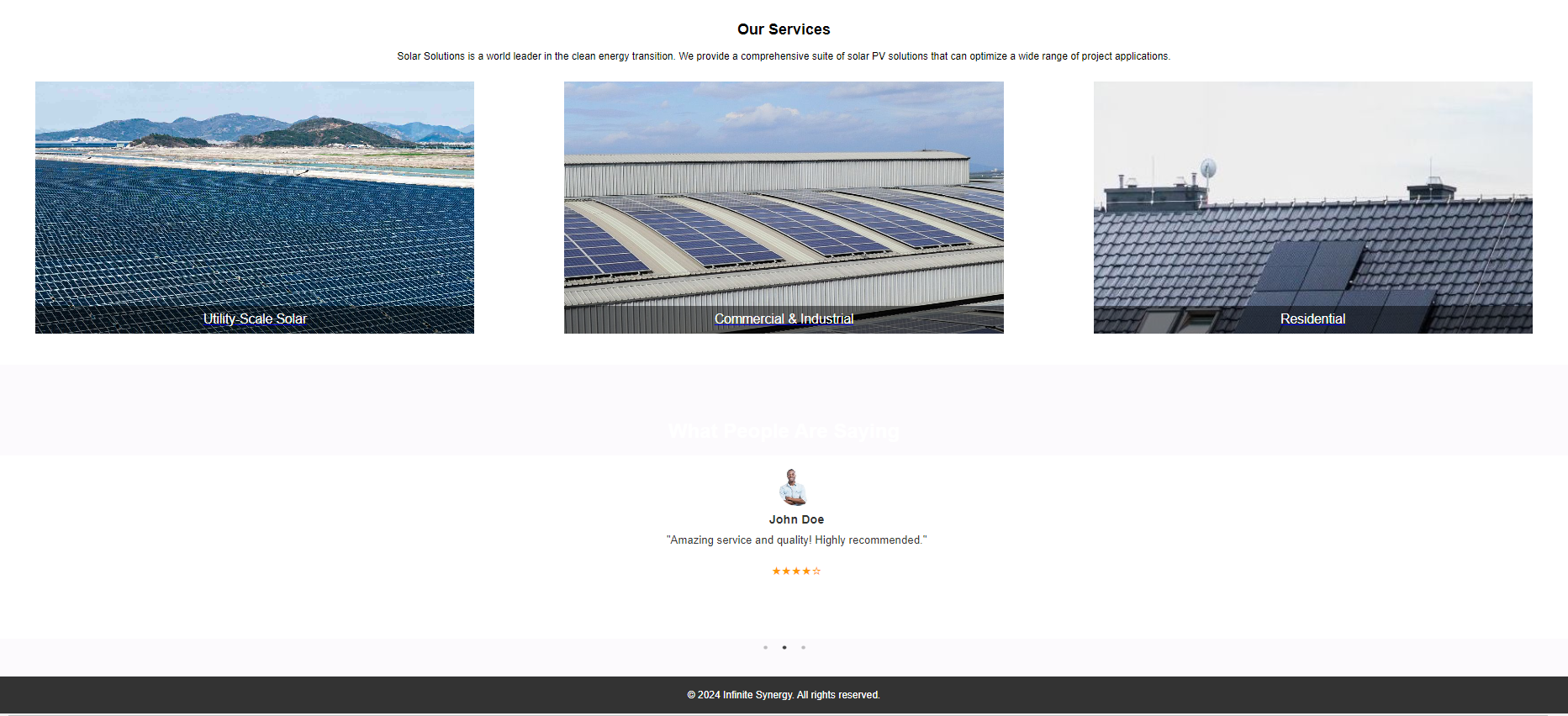
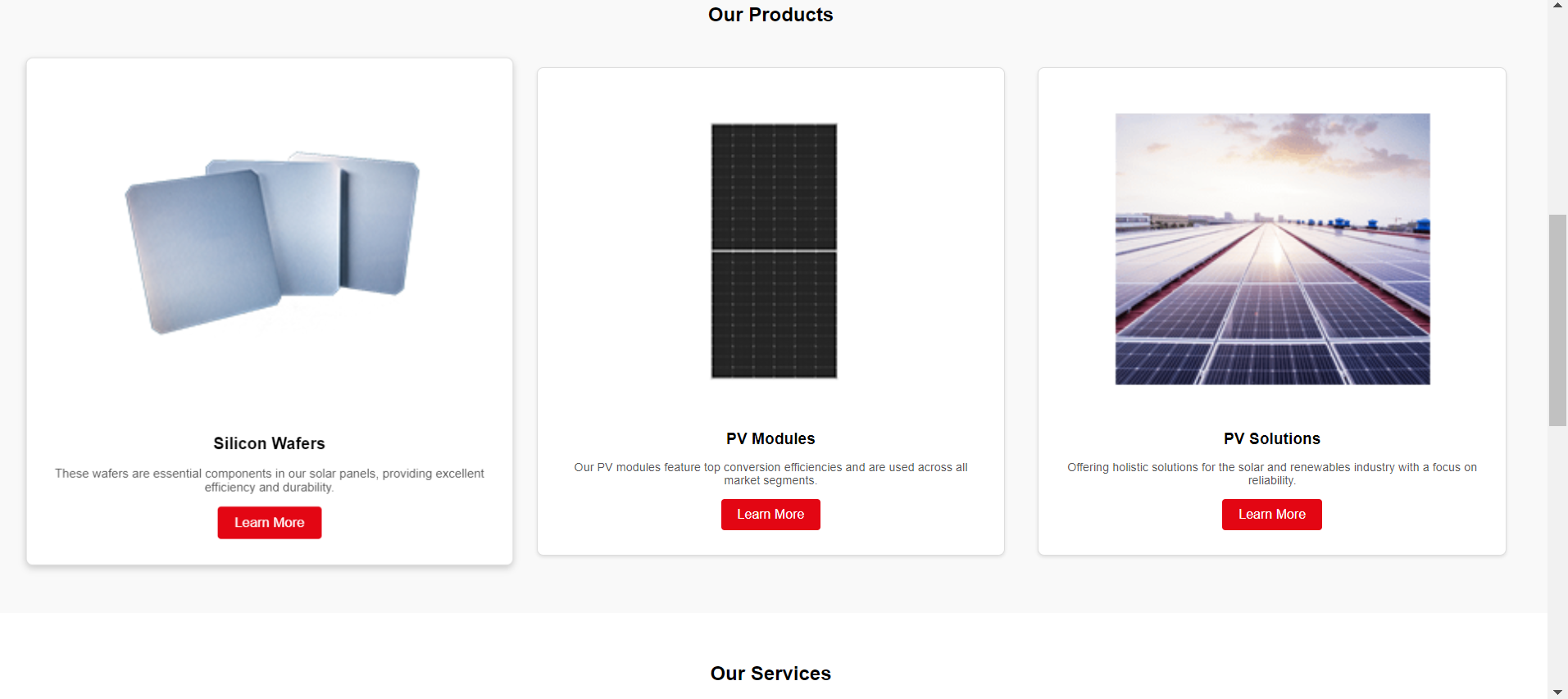


This opens the webpage, for frontend display. Won’t be functional until backend not enabled



*‘backend.sql*’ file required to create DB and table of *‘users’* with users’ data in it

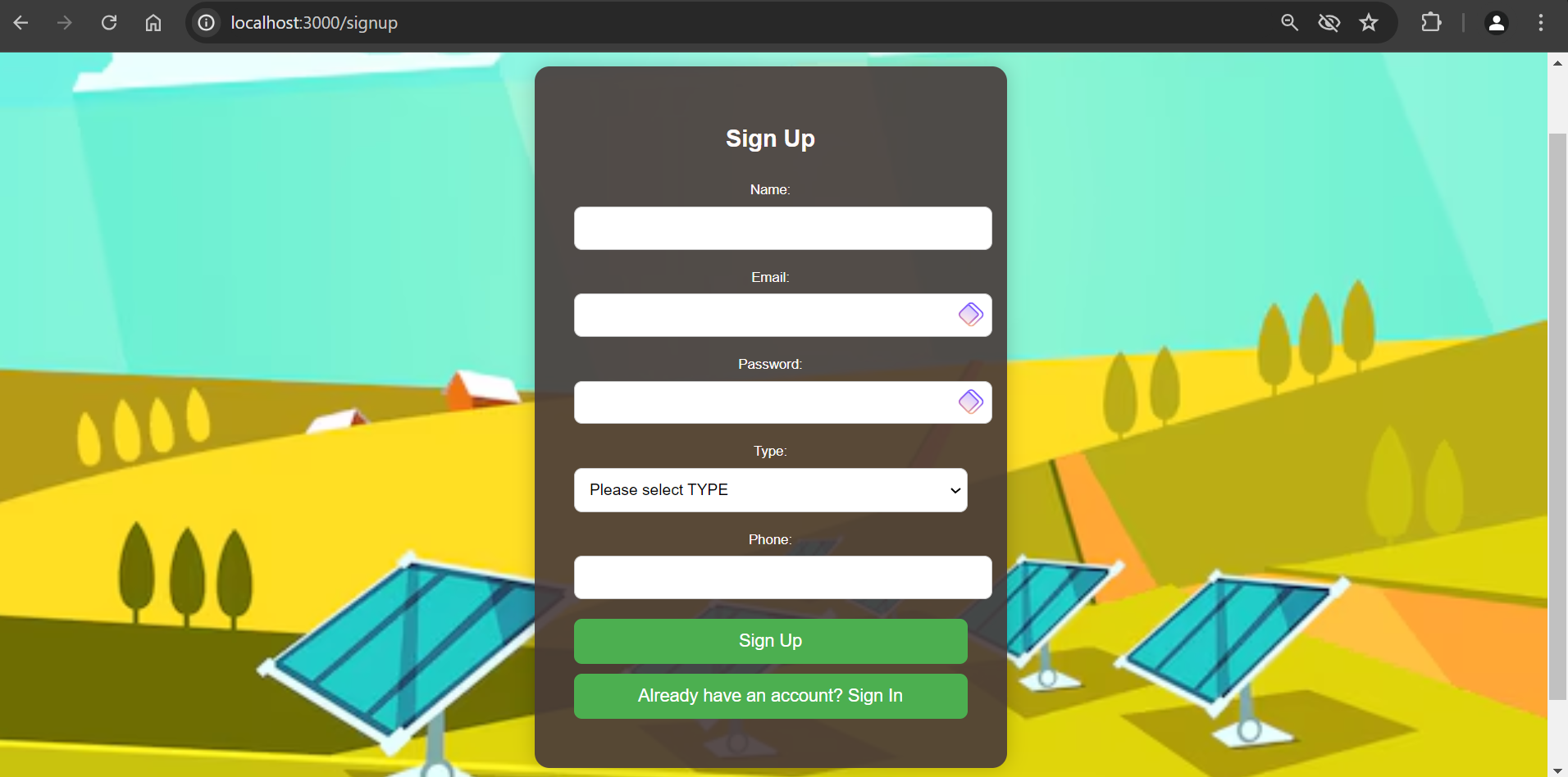


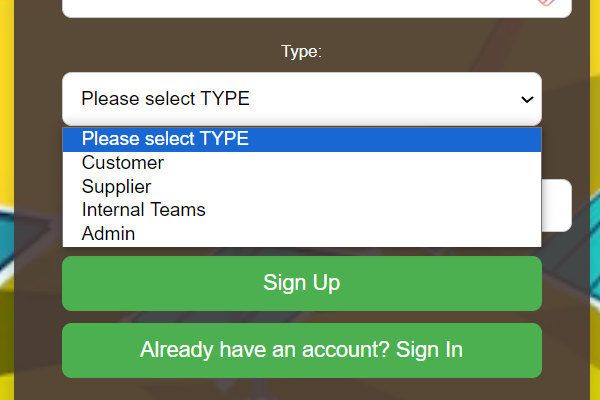


General display page for visiting users not have signed in

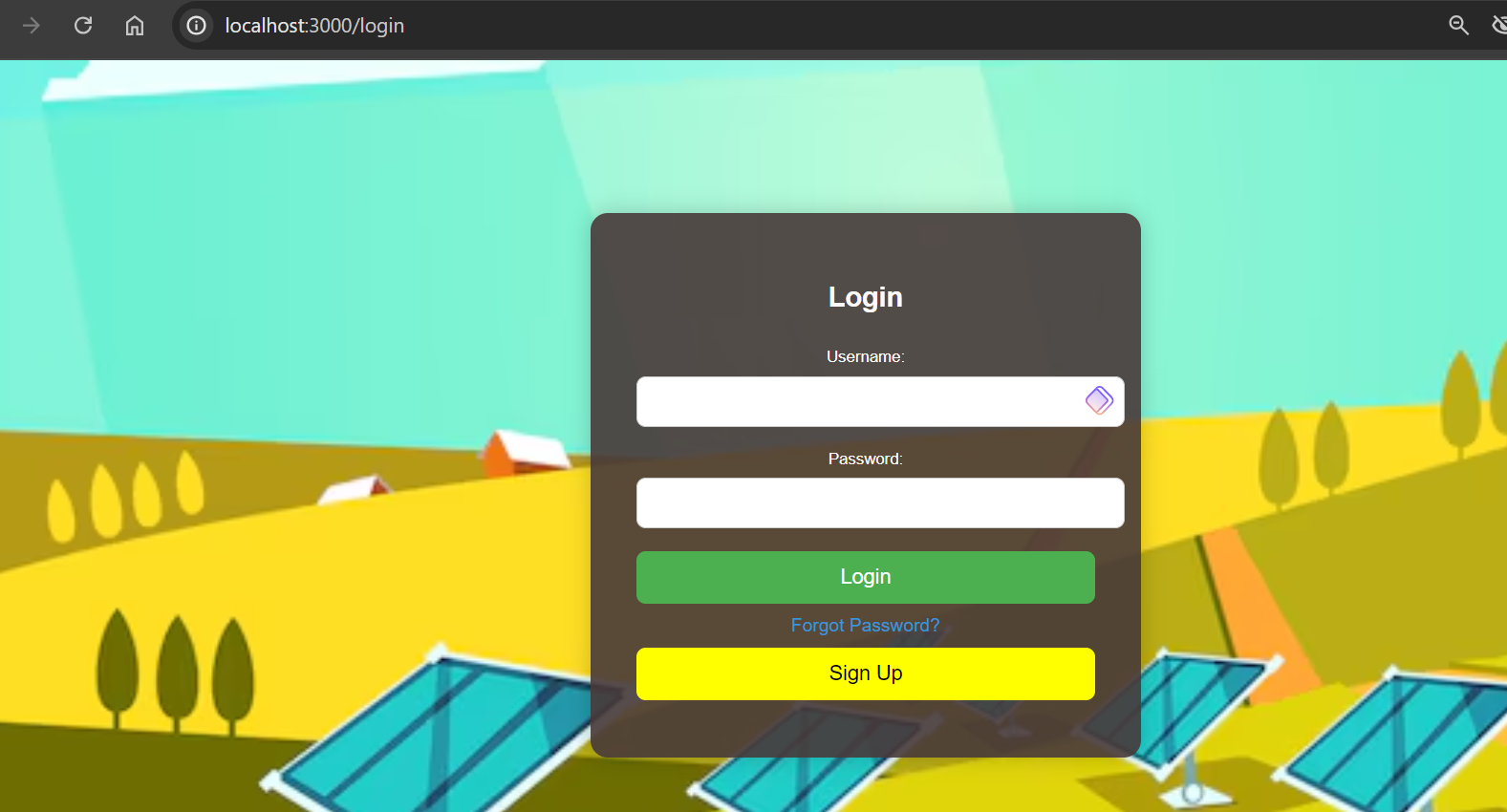


Functionality visible for only ’Suppliers’ i.e, supplier account has logged in. Upload product option, not yet connected through *cloudinary* to be functional





Fully operational signup page with DB + BCrypt Authentication integration, and error handling



Fully operational login page with DB + BCrypt Authentication integration, and error handling