Graphical user interface, application, timeline, Teams

Description automatically generated

**Part A: Set up fe & be**

**frontend**

react-bootstrap

Bootswatch – bootstrap.min.css

react-router-bootstrap

react-route-dom: router

aft set up frontend, init json pack in the root (make sure in root (neither fe or be)):

* npm init: in entry point, enter ‘server.js’
* npm i express

then **backend**:

1. server.js: set up route

in package.json in root: “scripts”: “start” : “node backend/server” – so that can run server with ‘npm start’ in terminal

then back to **frontend**:

in frontend folder: npm i axios (library to send http request from fe to be)

then in **root**:

npm I -D nodemon concurrently (dev dependencies)

then in root json package, in “scripts” add: "server": "nodemon backend/server",

"client": "npm start --prefix frontend",

“dev": "concurrently \"npm run server\" \"npm run client\""

Then in root terminal: npm run dev (now can run fe & be at the same time)

Still in **root**:

1. set up env file: all sensitive info (Paypal API etc) (make ensure .env files in .gitignore)

npm i dotenv

1. add ES modules:

In root json pack add after “main”:

"type": "module",

TAKEAWAYS:

1. bring any fe files to be, have to add .js

(e.g. in server:import products from './data/products.js')

**Part B: DB**

1. Section 4 mongoDB set up
2. Connect to DB:

.env

In root, npm i mongoose

Backend--> config --> db.js

Then bring it into server.js

1. set up schemas/ data models in backend- Modules
2. create sample data in backend – data

npm i bcryptjs

1. seeders.js

then in root json.package: add in “scripts”

"data:import":"node backend/seeder",

"data:destroy": "node backend/seeder -d",

Then in terminal: npm run data:import

1. set up routes 🡪 routes.js

npm i express-async-handler

1. set up postman
2. custom error handling: middleware

**Part C: Redux - frontend**

In frontend terminal: npm i redux react-redux redux-thunk redux-devtools-extension