C COMP 3123 Assignment – II Frontend (12%)

Docker Reference

Copy

The structure shown in the image is a project setup using Docker with separate folders for the front end and back end, along with a `docker-compose.yml` file at the root to manage container orchestration. Here's a summary of the setup for each component:

- **Root Directory (`studentID_comp3123_assignment/`):**
  - `docker-compose.yml`: The main Docker Compose file is used to define and manage services for both the frontend and backend applications, as well as any other services (like MongoDB) required by the app.
- **Frontend (`frontend/`):**
  - `Dockerfile`: Contains instructions to build a Docker image for the frontend React application.
  - `package.json`: Manages dependencies for the React front end.
  - `src/`: The source code folder for the React frontend application.
  - `public/`: The public directory for static assets in the React app.
- **Backend (`backend/`):**
  - `Dockerfile`: Contains instructions to build a Docker image for the backend Node.js application.
  - `package.json`: Manages dependencies for the Node.js backend.
  - `src/`: The source code folder for the backend application, including Express routes, controllers, and database models.

Example `docker-compose.yml` Setup

This file can orchestrate the frontend, backend, and MongoDB containers:

```yaml
version: '3'
services:
  frontend:
    build:
      context: ./frontend
      dockerfile: Dockerfile
    ports:
      - '3000:3000'
    depends_on:
      - backend
    environment:
      - REACT_APP_BACKEND_URL=http://backend:5000

  backend:
    build:
      context: ./backend
      dockerfile: Dockerfile
    ports:
      - '5000:5000'
    depends_on:
      - mongodb
    environment:
      - MONGO_URI=mongodb://mongodb:27017/mydatabase

  mongodb:
    image: mongo
    ports:
      - '27017:27017'
    volumes:
      - mongo-data:/data/db

  mongo-express:
    image: mongo-express:latest
    ports:
      - '8081:8081'
    environment:
      - ME_CONFIG_MONGODB_SERVER=mongodb
    depends_on:
      - mongodb

volumes:
  mongo-data:
```

Dockerfiles Overview

1. **Frontend Dockerfile (`frontend/Dockerfile`):**

   ```
   FROM node:14
   WORKDIR /app
   COPY package.json ./
   RUN npm install
   COPY . .
   CMD ["npm", "start"]
   ```

2. **Backend Dockerfile (`backend/Dockerfile`):**

   ```
   FROM node:14
   WORKDIR /app
   COPY package.json ./
   RUN npm install
   COPY . .
   CMD ["node", "src/index.js"]
   ```

With this structure, running `docker-compose up` in the **studentID_comp3123_assignment** directory will start all services, making it easier to manage and deploy the entire stack.

Previous
Submission

Last updated 1 year ago