**Full-Stack Todo List Application Documentation**

**Setup Instructions**

Prerequisites

* Docker & Docker Compose: Latest versions are recommended.
* Git: Required for cloning the repository.
* Terminal Skills: Basic command-line familiarity.

**Steps to Build, Run, and Stop Containers**

Clone the Repository:

* git clone <https://github.com/icnoka/fullstack-todo-list.git>cd fullstack-todo-list

Build and Run Containers:

* Run the following command to build Docker images and start all services:
* docker-compose up --build

This process will:

* Build the backend image from ./backend (exposing port 4000).
* Build the frontend image from ./frontend (exposing port 8000).
* Start a MongoDB container (exposing port 27017) with persistent storage.

Access the Application:

* Frontend: <http://localhost:8000>
* Backend API: [http://localhost:4000](http://localhost:4000/)
* MongoDB: Accessible via port 27017.

Stop Containers:

* To stop and remove containers, use: docker-compose down
* To also remove volumes, use: docker-compose down -v

**Network and Security Configurations**

Network Configurations

* Custom Network:  
   All services run on a custom bridge network named todonet, ensuring they can communicate seamlessly.

Exposed Ports:

* Frontend: Container port 8000 → Host port 8000.
* Backend: Container port 4000 → Host port 4000.
* MongoDB: Container port 27017 → Host port 27017.

**Security Settings**

Environment Variables:

Backend:

* NODE\_ENV=production ensures the app runs in production mode.
* MONGO\_URI=mongodb://mongodb:27017 connects to MongoDB using the service name.

Frontend:

* REACT\_APP\_API\_URL=http://localhost:4000 points the frontend to the backend API.

Database Credentials:

* For simplicity, explicit MongoDB credentials are not set. In production, add variables like MONGO\_INITDB\_ROOT\_USERNAME and MONGO\_INITDB\_ROOT\_PASSWORD, and update MONGO\_URI accordingly.

Volumes:

* Local code for both frontend and backend is mounted into their containers for live updates.
* A named volume (mongo\_data) is used for persisting MongoDB data.

**Troubleshooting Guide**

Common Issues and Resolutions

Frontend Not Loading:

* Verify the frontend container is running by accessing <http://localhost:8000>.
* Ensure the backend is operational at [http://localhost:4000](http://localhost:4000/).

Backend Errors:

* Check logs with: docker-compose logs backend
* Confirm that the MONGO\_URI is correctly configured and MongoDB is active.

MongoDB Connection Problems:

* Ensure no other service is using port 27017: docker ps
* If volume permission issues arise, adjust permissions (e.g., using chmod).

Build Failures:

Verify that the Node.js version (node:18.9.1) in the Docker files is compatible.

Rebuild images without cache if necessary: docker-compose build --no-cache