

User Manual for UKSHATI

A Next.js + MySQL Docker Project



Introduction

This project is a full-stack application built with **Next.js** for the frontend and **MySQL** for the database. It is containerized using **Docker** for easy setup and collaboration. The project is designed to provide a seamless development experience with pre-configured environment variables, database initialization, and Docker Compose setup.

Features

- **Frontend:** Next.js with API routes for backend logic.
- **Database:** MySQL with pre-configured schema and initialization.
- **Docker:** Containerized setup for seamless development.
- **Environment Variables:** Secure configuration using `.env` files.

Prerequisites

Before setting up the project, ensure you have the following installed:

- **Docker:** <https://docs.docker.com/get-docker/>
- **Docker Compose:** <https://docs.docker.com/compose/install/>
- **Node.js** (optional, for local development)

Project Structure

The project is structured as follows:

```

1 Ukshati/
2     frontend/                # Next.js application
3         pages/              # Next.js pages
4         public/            # Static assets
5         styles/            # CSS files
6         package.json       # Frontend dependencies
7         .env.local         # Environment variables for Next.js
8     db/                     # Database initialization scripts
9         company_db.sql     # SQL schema and data
10    docker-compose.yml      # Docker Compose configuration
11    Dockerfile              # Dockerfile for Next.js
12    .env.example           # Environment variables template
13    README.md              # Project documentation

```

Setup

1. Clone the Repository

Clone the repository to your local machine using the following command:

```

1 git clone <repository-url>
2 cd Ukshati

```

2. Set Up Environment Variables

1. Copy the `.env.example` file to `.env`:

```

1 cp .env.example .env
2

```

2. Update the `.env` file with your credentials. The file should look like this:

```

1 DB_HOST=db
2 DB_USER=company
3 DB_PASSWORD=Ukshati@123
4 DB_NAME=company_db
5 JWT_SECRET=HiUkshati123
6 DB_SSL=false
7 MYSQL_ROOT_PASSWORD=root
8 MYSQL_DATABASE=company_db
9 MYSQL_USER=company
10 MYSQL_PASSWORD=Ukshati@123
11

```

3. Start the Project

Run the following command to build and start the project:

```

1 docker-compose up --build

```

4. Access the Application

- **Frontend:** Open your browser and navigate to <http://localhost:3000>.
- **Database:** Accessible via db:3306 from the Next.js container.

Development Workflow

Run the Project

To start the project, use:

```
1 docker-compose up
```

Rebuild Containers

If you make changes to the Dockerfile or dependencies, rebuild the containers:

```
1 docker-compose up --build
```

Stop the Project

To stop the project, use:

```
1 docker-compose down
```

Remove Volumes and Containers

To remove volumes and containers, use:

```
1 docker-compose down -v
```

Access MySQL Database

To access the MySQL database, use:

```
1 docker exec -it <mysql-container-id> mysql -u company -pUkshati@123 company_db
```

Database Initialization

The database is initialized using the db/company_db.sql script. This script:

- Creates the employee table.
- Inserts sample data.

To reinitialize the database:

1. Stop the containers:

```
1 docker-compose down -v
2
```

2. Restart the project:

```
1 docker-compose up --build
2
```

Troubleshooting

1. npm Install Fails

- Ensure you have a stable internet connection.
- If behind a proxy, configure proxy settings in the Dockerfile:

```
1 npm install <package-name>
2
```

2. Database Connection Issues

- Verify the database is running:

```
1 docker ps
2
```

- Check MySQL logs:

```
1 docker logs <mysql-container-id>
2
```

3. Hot-Reload Not Working

Ensure the `volumes` section in `docker-compose.yml` is correctly configured:

```
1 volumes:
2   - ./frontend:/app
3   - /app/node_modules
```

Contributing

To contribute to the project:

1. Fork the repository.
2. Create a new branch:

```
1 git checkout -b feature/your-feature
2
```

3. Commit your changes:

```
1 git commit -m "Add your feature"
2
```

4. Push to the branch:

```
1 git push origin feature/your-feature
2
```

5. Open a pull request.

Support

For support or questions, contact the maintainer at jaideepn3590@duck.com.

Conclusion

This manual provides a comprehensive guide to setting up, using, and contributing to the **Next.js + MySQL Docker Project**. For additional details, refer to the `README.md` file in the repository.