

## # Deployment Assistance Guide

This guide explains how to set up and deploy the GST Automation application for both local and hosted environments.

### ## 1. Prerequisites

Ensure the following software is installed on your system:

- **Node.js** (v16 or higher) - [Download Node.js](https://nodejs.org/)
- **MongoDB** - [Download MongoDB](https://www.mongodb.com/try/download/community)
- **Docker** (Optional, for containerized deployment) - [Download Docker](https://www.docker.com/products/docker-desktop)
- **Git** (Optional, for version control) - [Download Git](https://git-scm.com/)

### ## 2. Setting Up Locally

#### ### 2.1 Frontend

1. Navigate to the `frontend` directory:

```
```bash
cd frontend
...`
```

2. Install dependencies:

```
```bash
npm install
...`
```

3. Create an `.env` file based on the `.env.example`:

```
```bash
```

```
cp .env.example .env
```

```
...
```

4. Update the backend API URL in `.env` if required.

5. Start the frontend development server:

```
```bash
```

```
npm start
```

```
...
```

### ### 2.2 Backend

1. Navigate to the `backend` directory:

```
```bash
```

```
cd backend
```

```
...
```

2. Install dependencies:

```
```bash
```

```
npm install
```

```
...
```

3. Create an `.env` file based on the `.env.example`:

```
```bash
```

```
cp .env.example .env
```

```
...
```

4. Update environment variables for MongoDB, Razorpay, and JWT as needed.

5. Start the backend server:

```
```bash
```

```
npm start
```

```
...
```

### ### 2.3 Database

1. Ensure MongoDB is running locally or connect to a cloud MongoDB instance.
2. Update the `MONGO\_URI` in the backend `.env` file with your MongoDB connection string.

## ## 3. Docker-Based Deployment

### ### 3.1 Building the Images

1. Navigate to the root directory containing the `docker-compose.yml` file.
2. Build and start the containers:

```
```bash
docker-compose up --build
```
```

### ### 3.2 Access the Application

- Frontend: [http://localhost:3000](http://localhost:3000)
- Backend API: [http://localhost:5000/api](http://localhost:5000/api)
- MongoDB: Runs on port `27017`.

### ### 3.3 Stop the Containers

To stop the Docker containers, run:

```
```bash
docker-compose down
```
```

## ## 4. Hosted Deployment

### ### 4.1 Steps for Deployment

### 1. **\*\*Frontend\*\***:

- Build the React application:

```
```bash
```

```
npm run build
```

```
```
```

- Deploy the build folder to a hosting service like AWS S3, Netlify, or Vercel.

### 2. **\*\*Backend\*\***:

- Deploy the Node.js application to a cloud provider like AWS, Azure, or Heroku.
- Ensure the `.env`` file is correctly configured in the hosting environment.

### 3. **\*\*Database\*\***:

- Use a managed MongoDB service like MongoDB Atlas or host your own.

## ## 5. Notes

- Ensure sensitive keys (e.g., Razorpay, JWT secret) are securely stored in environment variables.
- Test the application thoroughly after deployment.