**AgroMart - Enabling Farming Product Online**

**Description:**

AgroMart is a platform that connects farmers and allows them to buy agricultural products directly for farming. The platform works to provide fair prices for farmers while providing customers with fresh, high-quality product.

==================================================================

# Instruction for Creating Database on MongoDB

1. **Open the MongoDB Compass:** Start by running MongoDB Compass on your system. If you haven't installed it yet, you can get it from the MongoDB website. Download the stable version of MongoDB Compass GUI.
2. **Connect to MongoDB Server:** After launching MongoDB Compass, you will be prompted to connect to a MongoDB server. Click the "Connect" button.
3. **Go to the Databases tab:** Once connected you'll see a list of databases in the left sidebar. If you don't see any databases, that means you haven't created any yet. To manage databases, select the "Databases" tab.
4. **Create a New Database:** To start a new database, click the "Create Database" button.
5. **Enter Database Name:** A dialog box will appear asking you to enter the name of the new database. Type in "agromart" name for database.
6. **Confirm Creation:** After entering the database name and configuring any optional settings, click on the "Create" button to create the new database.
7. **Verify:** You should now see your newly created database name in the left sidebar under the "Databases" tab. Import all the provided database JSON file on it.

# Instruction for Running Backend Code

1. **Open the Backend code:** Start by opening all code on VS Code and navigate to the directory where backend code is located.
2. **Change directory to Backend:** Open a terminal window and navigate to the directory containing backend code. You can use the cd command followed by the path to the directory.
3. **Install required packages:** Before running the server, need to install required packages. If project has a package.json file with a list of packages, you can simply run: npm install

This command will install all the packages listed in the package.json file.

1. **Command to run server:** Once the required packages are installed, you can start your backend server with the command ‘nodemon’.

# Instruction for Running Dashboard WebApp code

1. **Open the Dashboard WebApp code:** Start by opening all code on VS Code and navigate to the directory where Dashboard WebApp code located.
2. **Change directory to Dashboard WebApp:** Open a terminal window and navigate to the directory containing Dashboard WebApp code. You can use the cd command followed by the path to the directory.
3. **Install required packages:** Before running the Dashboard WebApp, need to install required packages. If project has a package.json file with a list of packages, you can simply run: npm install

This command will install all the packages listed in the package.json file.

1. **Command to run Dashboard WebApp:** Once the required packages are installed, you can start Dashboard WebApp server with the command:

‘npm start’.

# Instruction for Running Customer Mobile App code

1. **Open the Customer Mobile App code:** Start by opening code on VS Code and navigate to the directory where Customer MobileApp code located.
2. **Change directory to Customer MobileApp:** Open a terminal window and navigate to the directory containing Customer MobileApp code. You can use the cd command followed by the path to the directory.
3. **Install required packages:** Before running the Customer MobileApp, need to install required packages. If project has a package.json file with a list of packages, you can simply run: npm install

This command will install all the packages listed in the package.json file.

1. **Emulator:** Open Emulator from the Android Studio.
2. **Command to run Customer MobileApp:** Once the required packages are installed and Emulator open, you can start Customer MobileApp server with the command:

‘npx expo start’.