

## AIDAR Connect *Setup* guide

---

If you want to directly setup the project from github to vscode check the bottom section. (No need to read through all these documentation – These are to make the project from scratch).

---

### Set Up MongoDB Atlas (Use mine; Already setup in codebase - .env file)

#### 1. Create a MongoDB Atlas Account:

- Setup MongoDB Atlas account from the website.

#### 2. Create a Cluster:

- Create a new project and build a cluster. Choose the closest region to reduce latency.

#### 3. Configure Database Access:

- Go to **Database Access** and create a new database user. Set a username and password, and make sure to note them down as you'll need them later.

#### 4. Allow Network Access:

- Go to **Network Access** and allow access from anywhere by adding 0.0.0.0/0 to the IP whitelist. You can restrict it to specific IPs for security if needed.

#### 5. Get the Connection String:

- Go to **Clusters**, click **Connect**, and select **Connect Your Application**. Copy the connection string, which will look like this:

```
mongodb+srv://<username>:<password>@cluster0.mongodb.net/<dbname>?retryWrites=true&w=majority
```

---

## Set Up the Backend (Node.js and Express)

### 1. Initialize the Project:

- Create a new directory for your project and navigate into it:
- Initialize a Node.js project: `npm init -y`

### 2. Install Dependencies:

- Install Express, Mongoose (MongoDB ORM), dotenv (for environment variables), and CORS:

```
npm install express mongoose dotenv cors
```

```
npm install --save-dev nodemon
```

### 3. Set Up Folder Structure:

- Organize the backend folder structure:

### 4. Configure MongoDB Connection:

- In the config folder, create `db.js` for MongoDB configuration:

### 5. Set Up Models:

- Create a Mongoose model for surveys in the models folder:

### 6. Define API Routes:

- Create `createRoutes.js` in the routes folder:

### 7. Set Up the Server:

- In the root directory, create `server.js` as the entry point for the server:

### 8. Add Environment Variables:

- Create a `.env` file in the root directory:

### 9. Run the Server:

- Start the server: `node server.js`
-

## Set Up the Frontend (React)

### 1. Create the React App

- In the root of the project directory, create a new React app in a subfolder client:
  - `npx create-react-app client`

### 2. Install Axios

- Inside the client directory, install Axios, used for making requests to the backend API: `npm install axios`

### 5. Configure a Proxy for API Requests

- To avoid specifying the server's full URL (`http://localhost:5000`) in Axios calls, you can set up a proxy in the client package.
- In the `client/package.json` file, add this line: `"proxy": "http://localhost:5500"`

### 6. Start the React App

- Start the React development server by running the following command in the client directory: `npm start`

---

### Dependencies for server:

- Cors
- Dotenv
- Express
- Mongoose
- Nodemon

### Dependencies for client:

- @Popperjs/core
- Axios
- Bootstrap
- Bootstrap-icons
- React-beautiful-dnd
- React-bootstrap
- React-bootstrap-icons
- React-icons
- React-quill

---

## Steps to setup project from Github repository directly into VSCode:

### 1. Clone from Github repository using source control

- Go to source control on VSCode
- Click clone a repository
- Enter URL - <https://github.com/agupt001/aidar-survey-builder-app.git>
- Select Project destination

### 2. Go to integrated terminal of VSCode to install dependencies

- Click Terminal -> New Terminal
- Change directory: `cd aidar-connect/`
- Install nodeJS and npm (if you don't already have it)
  - Linux: `sudo apt install nodejs npm`
  - Windows/ Mac: Download from official website and install
- Install server side dependencies
  - `npm install`
- Install client side dependencies
  - `cd client/`
  - `npm install`

### 3. Run server

- Open a terminal
- `cd aidar-connect/`
- `node server.js`

### 4. Run client

- Open a terminal
- `cd aidar-connect/client/`
- `npm start`

### 5. Client server will start at - `http://localhost:3000` or `http://127.0.0.1:3000`