# **Frontend Setup Guide for fbAutoClient**

This guide will walk you through setting up the frontend of the fbAutoClient project on your local machine. Follow the steps carefully to ensure everything works smoothly.

## **Prerequisites**

Before starting, make sure you have the following installed on your system:

* [Node.js](https://nodejs.org/) (LTS version recommended, e.g., 18.x or higher)  
  Node.js comes with npm (Node Package Manager), which is required to install project dependencies.
* Git (if cloning the repo instead of downloading a ZIP)  
  You can check if Git is installed by running:  
  git --version

## **Step 1: Get the Project Code**

You can either clone the repository or download the ZIP file.

### **Option A: Clone via Git**

git clone https://github.com/Prudvi0033/fbAutoClient.git

cd fbAutoClient

### **Option B: Download ZIP**

1. Ge the zip file from the drive it self download it.
2. Extract the ZIP file into a folder (e.g., Documents/fbAutoClient).
3. Navigate into that folder:  
   cd fbAutoClient

## **Step 2: Create Environment File**

Inside the project root directory, create a new file named .env.  
Add the following line to configure the API endpoint:

NEXT\_PUBLIC\_API\_URL = “YOUR BACKEND URL”

### **Why is this needed?**

* The frontend is built with Next.js, and it uses environment variables to communicate with the backend.
* The prefix NEXT\_PUBLIC\_ ensures this variable is accessible in the browser as well.
* This URL points your frontend to the backend API hosted on Render.

## **Step 3: Install Dependencies**

Run the following command inside the project folder:

npm install

This will download and set up all required packages defined in package.json.

## **Step 4: Start the Development Server**

Once installation is complete, start the Next.js development server:

npm run dev

By default, the app will be available at:

[http://localhost:3000](http://localhost:3000/)

## **Common Issues & Fixes**

* npm: command not found  
  → Install Node.js again and ensure it’s added to your PATH.
* Port 3000 already in use  
  → Either stop the process using port 3000 or run:  
  npm run dev -- -p 4000  
  (this will start the app on port 4000).

## **You’re All Set!**

You now have the frontend of fbAutoClient running locally. You can start building and testing features immediately.

# **Backend Setup Guide for fbAuto**

This guide explains how to set up and run the fbAuto backend project on your local machine. Follow the steps carefully to ensure everything is working as expected.

## **Prerequisites**

Before you begin, make sure you have installed:

* [Node.js](https://nodejs.org/) (LTS version recommended, e.g., 18.x or higher)
* npm (comes bundled with Node.js)
* Git (to clone the repo, unless you download a ZIP)
* PostgreSQL Database (a connection is already provided via Neon in .env)

## **Step 1: Get the Project Code**

You can either clone the repository or download the ZIP.

### **Option A: Clone via Git**

git clone https://github.com/Prudvi0033/fbAuto.git

cd fbAuto

### **Option B: Download ZIP**

1. Download the zip file from the drive itself
2. Extract the ZIP file (e.g., to Documents/fbAuto).
3. Navigate into that folder:  
   cd fbAuto

## **Step 2: Create Environment File**

Inside the project root directory, create a file named .env.  
Paste the following environment variables:

# Runner options

HEADLESS=true

SLOWMO=500

DEBUG=true

# Server config

PORT=5000

NODE\_ENV=development

JWT\_SECRET=hrdashboard

# Database

DATABASE\_URL='YOUR NEON DB INSTANCE'

# External services

N8N\_JOB\_CONTEXT\_WEBHOOK\_URL = ‘YOUR WEBHOOK URL’

DOMAIN\_URL = ‘BACKEND URL’

### **Explanation:**

* DATABASE\_URL: PostgreSQL database hosted on Neon.
* JWT\_SECRET: Used for authentication & token signing.
* HEADLESS / SLOWMO / DEBUG: Runner options for automation/debugging.
* FACEBOOK\_VERIFY\_TOKEN: Token required for Facebook webhook verification.
* N8N\_JOB\_CONTEXT\_WEBHOOK\_URL: Webhook URL for job automation (n8n).
* DOMAIN\_URL: Public domain of your backend API.

## **Step 3: Install Dependencies**

Install the required dependencies for both the root and the server folder.

npm i && cd server && npm i

## **Step 4: Prisma Setup**

Generate the Prisma client inside the server folder:

cd server

npx prisma generate

This ensures Prisma can interact with your PostgreSQL database.

## **Step 5: Start the Backend Server**

### **Option A: Run with Global Dev Script**

npm run dev

### **Option B: Run Manually from Server Folder**

cd server

node src/index.js

By default, the server runs on:  
👉 [http://localhost:5000](http://localhost:5000/)

## **Common Issues & Fixes**

* Database connection error  
  → Ensure the DATABASE\_URL in .env is correct and that your Neon database is accessible.

Prisma errors (client not found)  
→ Run:  
cd server

* npx prisma generate
* Port already in use (5000)  
  → Stop the process using port 5000, or update the PORT value in .env.

## **You’re All Set!**

You now have the fbAuto backend running locally. With both the backend and frontend set up, you’re ready to develop and test the application.