**DEBUGGING**

Perfect! Since you have **pgAdmin 4**, you can complete **Step 2 (Database Setup)** fully through the pgAdmin GUI. Here's how to proceed step by step:

## ✅ STEP 2 — PostgreSQL Setup Using pgAdmin 4

### 🔹 2.1: Open pgAdmin 4

Launch **pgAdmin 4**.

Enter the password you set during PostgreSQL installation.

### 🔹 2.2: Create a New Database

In the left panel, expand:

Servers → PostgreSQL <version> → Databases

Right-click on **Databases** → Click **Create → Database...**

Fill:

**Database name**: bel\_project

Leave owner as default (postgres)

Click **Save**

✅ You now have an empty bel\_project database ready.

### 🔹 2.3: Create users Table

Expand your new DB:

Servers → PostgreSQL → Databases → bel\_project → Schemas → public

Right-click on **Tables** → Click **Query Tool**

Paste this SQL code:

CREATE TABLE users (

id SERIAL PRIMARY KEY,

username VARCHAR(50) NOT NULL,

staff\_id VARCHAR(20) UNIQUE NOT NULL,

password VARCHAR(200) NOT NULL

);

Click the **Execute/Run (▶️) button** at the top.

✅ You now have a users table inside the bel\_project database.

## ✅ 3. How to ****check**** if data is stored in PostgreSQL?

You can **open pgAdmin 4**:

### Steps in pgAdmin:

Open pgAdmin 4

Go to:

Servers > PostgreSQL > Databases > bel\_project > Schemas > public > Tables > users

Right-click on users table → **View/Edit Data** → **All Rows**

You’ll see entries like:

| **id** | **username** | **staff\_id** | **password** |
| --- | --- | --- | --- |
| 1 | shashi | 123456 | $2b$10$... |

This confirms that your **Register flow works** ✅

1. Now extract the new zip file of ur project and run commands normally after uninstall old versionof nodejs(page 13) installing new version of nodejs, python and vs code installers.(page 15)
2. if above step fails follow step( page 10)

when i give npm -v version is shown  
ans:  
Great — if npm -v shows a version number, that means **Node.js and npm are installed correctly** on your system ✅.

But your error is not about npm — it's about react-scripts missing. That package is **not installed** in your frontend directory.

:

rm -rf node\_modules package-lock.json # For Windows use Git Bash or PowerShell

npm install

npm start

If you can’t run rm -rf in Windows CMD, do this manually:

**Delete** the node\_modules folder.

**Delete** the package-lock.json file.

Then run:

npm install

npm start

### Need Help?

If you share your package.json file from the frontend/ folder, I can double-check if react-scripts is present.

Would you like me to check that for you?

i think react scrpt is present  
ans:  
Great — let’s confirm that quickly.

### ✅ To verify if react-scripts is present:

Open the file:

C:\Users\Administrator\Desktop\BEL-project-management-master-new\BEL-project-management-master\frontend\package.json

Look under the "dependencies" section. You should see something like:

"dependencies": {

"react": "^18.2.0",

"react-dom": "^18.2.0",

"react-scripts": "5.0.1",

...

}

✅ If react-scripts is there, you’re good.

### 🚀 Now, Run These Commands

Open **Command Prompt** or **PowerShell** and navigate to the frontend folder:

cd "C:\Users\Administrator\Desktop\BEL-project-management-master-new\BEL-project-management-master\frontend"

Then:

npm install

npm start

### 🛠 If you still face issues like:

react-scripts: command not found

Then install it directly:

npm install react-scripts --save

Then try:

npm start

Would you like me to check the package.json if you upload it here?

it shows npm ERR! and something  
ans:  
Got it — since you're still seeing npm ERR!, that means something is going wrong during setup. Let’s fix it step by step.

### 🔍 First, let’s ****diagnose the problem clearly****:

Please **upload your** package.json **file** from the frontend folder:

C:\Users\Administrator\Desktop\BEL-project-management-master-new\BEL-project-management-master\frontend\package.json

That will help me:

Verify whether react-scripts is included or missing.

Confirm the correct versions of dependencies.

### 🔧 In the meantime, try these commands:

Make sure you're in the frontend folder, then run:

rm -rf node\_modules package-lock.json

(if you're on Windows CMD, instead run:)

rmdir /s /q node\_modules

del package-lock.json

Then:

npm cache clean --force

npm install

npm start

### ✅ What we’re trying to ensure:

node\_modules is rebuilt cleanly.

react-scripts is installed correctly.

No cached/broken install is interfering.

Please upload the package.json or paste its contents here — then I can give you an exact fix.

----------------------------------------------------------------

can i run npm install in offline system  
Great question — here's a **clear answer**:

## ❌ You cannot run npm install in an offline system by default

### Why?

When you run:

npm install

NPM:

Fetches packages from the internet (registry.npmjs.org)

Even if package.json exists, without internet it **cannot download** dependencies

## ✅ Solution: Pre-install dependencies online, then transfer to offline system

You have **2 options**:

### ✅ ****Option 1: Include**** node\_modules ****in your ZIP****

On your **online system**, go to both frontend and backend, and run:

npm install

This will generate a complete node\_modules folder in each.

Now zip the whole project **with the** node\_modules **folders included**:

project/

├── backend/

│ ├── package.json

│ ├── node\_modules/ ✅

├── frontend/

│ ├── package.json

│ ├── node\_modules/ ✅

Transfer this ZIP to the offline system.

✅ Now you can directly run:

node index.js in backend

npm start in frontend  
**without running** npm install

### ✅ ****Option 2: Use**** npm pack ****for true offline installation**** (advanced)

This is helpful if you **want to run** npm install **offline**, but requires effort:

On your online system, for both frontend and backend:

npm install

npm pack

This generates a .tgz tarball like your-project-1.0.0.tgz.

Copy the .tgz file to the offline system and run:

npm install ./your-project-1.0.0.tgz

This will install the package offline using the tarball.

steps to setup the project offline using frontend and backedn .tgz

Ans:  
  
Here's a **complete step-by-step guide** to set up your **BEL Project (frontend and backend)** **offline** using the .tgz files you created with npm pack.

## 🧩 STEP-BY-STEP INSTALLATION (Offline)

Let’s say you've transferred:

frontend-x.x.x.tgz (e.g. frontend-1.0.0.tgz)

backend-x.x.x.tgz (e.g. backend-1.0.0.tgz)

Your original frontend and backend folder **(without node\_modules)**

### 📁 1. Extract Project Folder

Create a folder, e.g., bel\_project\_offline

Place your project source code in this folder (both frontend and backend)

Place the .tgz files inside their respective folders.

### 📦 2. Install Frontend Dependencies from .tgz

Open Command Prompt.

Navigate to frontend:

cd path\to\bel\_project\_offline\frontend

Install from .tgz:

npm install ./frontend-1.0.0.tgz

(Replace filename if different.)

Start frontend:

npm start

### ⚙️ 3. Install Backend Dependencies from .tgz

Open **another** Command Prompt window.

Navigate to backend:

cd path\to\bel\_project\_offline\backend

Install from .tgz:

npm install ./backend-1.0.0.tgz

Start backend:

node index.js

(Or your main file, if different.)

### ✅ Notes

**You do not need internet.**

This installs dependencies from the packed .tgz instead of the npm registry.

If .env or PostgreSQL setup is needed, make sure to replicate those settings too.

uninstall and install npm and node of correct version  
  
ans:  
Sure! Here's everything you need to:

## ✅ ****1. Download Node.js v18.18.2 + NPM v9.8.1 (Offline Installer)****

**🔗 Official MSI (Windows Installer):**

[Download Node.js v18.18.2 (64-bit .msi)](https://nodejs.org/download/release/v18.18.2/node-v18.18.2-x64.msi)

🟢 This installer also installs:

Node.js v18.18.2 ✅

NPM v9.8.1 ✅ (comes bundled with this Node version)

Once downloaded, copy this .msi file into your offline system (via USB or any medium).

## 🔁 ****2. Steps to Uninstall Current Node.js + NPM in Offline System****

### 💻 On Windows:

Press Windows + R, type appwiz.cpl, and hit Enter.

Look for **Node.js** in the list of installed programs.

Right-click it → **Uninstall**.

Wait for complete removal.

🔁 This will uninstall both **Node** and **NPM**.

## 🛠️ ****3. Install Node.js v18.18.2 on Offline System****

Copy the .msi file (from Step 1) to the offline system.

Double-click it to launch the setup.

Follow the steps → click **Next**, **Accept**, and **Install**.

After installation, open **Command Prompt** and verify:

node -v

# should show: v18.18.2

npm -v

# should show: 9.8.1

## ✅ 4. (Optional) NPM Manual Check:

Even if you're offline, you can check:

where node

where npm

This confirms both are in C:\Program Files\nodejs\.

Python and other software setup:  
ans:  
Here are the **official download links** and **offline setup steps** to prepare your system for running bcrypt or any package that needs native build tools like node-gyp.

## ✅ ONLINE SYSTEM – Download Links

### 1. ****Python 3.x (for**** node-gyp****)****

🔗 [Download Python 3.11.9 (Windows installer)](https://www.python.org/ftp/python/3.11.9/python-3.11.9-amd64.exe)

✅ Make sure to check **“Add Python to PATH”** during installation.

### 2. ****Microsoft Visual Studio Build Tools****

🔗 [Download Build Tools for Visual Studio 2022](https://aka.ms/vs/17/release/vs_BuildTools.exe)

During installation, select:

**"C++ build tools"**

Ensure the **"Windows 10 SDK"** or **"Windows 11 SDK"** is checked.

### 3. ****.NET Framework 4.5+****

Most Windows 10/11 machines already have this. But if needed:

🔗 [Download .NET Framework 4.8 Runtime (Offline Installer)](https://go.microsoft.com/fwlink/?linkid=2088631)

## ✅ OFFLINE SYSTEM – Installation Steps

### 🧳 Transfer to USB or External Drive

Copy the following into a folder like bel-offline-setup:

bel-offline-setup/

├── python-3.11.9-amd64.exe

├── vs\_BuildTools.exe

├── ndp48-x86-x64-allos-enu.exe

└── (your frontend & backend .tgz files)

## 🖥️ STEPS TO INSTALL ON OFFLINE MACHINE

### 1. ****Install Python 3.11****

Run python-3.11.9-amd64.exe

Make sure **“Add Python to PATH”** is checked.

Verify with:

python --version

### 2. ****Install Visual Studio Build Tools****

Run vs\_BuildTools.exe

Select:

✅ C++ Build Tools

✅ Windows 10/11 SDK

Wait for installation (can take a while)

### 3. ****Install .NET Framework****

Run ndp48-x86-x64-allos-enu.exe

Restart if prompted

### 4. ****Install Frontend and Backend****

Assuming:

You’ve already installed Node.js v18.18.2 (from node-v18.18.2-x64.msi)

You transferred the .tgz files

Then:

# Example for frontend

cd path\to\frontend

npm install ./frontend-x.x.x.tgz

# Example for backend

cd path\to\backend

npm install ./backend-x.x.x.tgz

# Then run both

npm start # frontend

node index.js # backend

Would you like me to create a **checklist PDF** of these steps for print or distribution?

Error Breakdown:

npm : File C:\Program Files\nodejs\npm.ps1 cannot be loaded because running scripts is disabled on this system.

PowerShell is blocking the script npm.ps1 because the execution policy is set to restrict running scripts.

---

💻 How to Fix It (Just follow these steps):

1. Open PowerShell as Administrator

Click Start

Type PowerShell

Right-click it and click "Run as administrator"

2. Run this command (Copy-paste it):

Set-ExecutionPolicy RemoteSigned

3. It will ask you for confirmation:

Type Y and press Enter

---

What this does:

RemoteSigned allows local scripts to run (like your npm start) but still protects you from risky downloaded scripts.

---

🛠 Now Try Again:

After changing the policy:

1. Close the old terminal

2. Reopen VS Code terminal

3. Run:

cd frontend

npm start

## 💾 Official Download Links

Here are the official URLs you should add to your Word doc. Just write them like:

### 1. ****Visual Studio Build Tools****

🔗 Download Visual Studio Build Tools  
https://aka.ms/vs/17/release/vs\_BuildTools.exe

✅ This is the direct .exe link from Microsoft for the latest Build Tools.

### 2. ****Python 3.11.9 (64-bit)****

🔗 [Download Python 3.11.9 (64-bit)](https://www.python.org/ftp/python/3.11.9/python-3.11.9-amd64.exe)

✅ Direct from Python.org — latest stable 3.11.9 version.

### 3. ****Node.js v18.18.2 (64-bit)****

🔗 Download Node.js v18.18.2 (64-bit)

✅ Official MSI installer from nodejs.org (long-term support version).

### 4. ****.NET Framework 4.8****

🔗 Download .NET Framework 4.8 Runtime

✅ From Microsoft — full installer for all OS versions.