

How to Add These Dependencies to Your GitHub Project

To integrate the provided list of dependencies into your project, follow the steps below. This process will ensure that your GitHub repository is up to date and that any collaborator can install the exact same dependencies for development or deployment.

1. Prepare Your Environment

- Confirm you have **Node.js** and **npm** (Node Package Manager) installed.

You can verify by running:

```
node -v
npm -v
```

2. Create or Update `package.json`

If your project doesn't have a `package.json` file yet, initialize it with:

```
npm init -y
```

3. Install Dependencies

A. Batch Install via Command Line

You can install all dependencies at once by running:

```
npm install \
@googlemaps/js-api-loader@^1.16.10 \
@radix-ui/react-avatar@^1.1.10 \
@radix-ui/react-checkbox@1.1.3 \
@radix-ui/react-dialog@1.1.5 \
@radix-ui/react-label@2.1.1 \
@radix-ui/react-popover@1.1.5 \
@radix-ui/react-progress@1.1.1 \
@radix-ui/react-select@2.1.5 \
@radix-ui/react-slider@1.2.2 \
@radix-ui/react-slot@1.1.1 \
@radix-ui/react-switch@1.1.2 \
@radix-ui/react-toggle@1.1.1 \
@radix-ui/react-tooltip@1.1.7 \
@types/bcryptjs@^2.4.6 \
@types/better-sqlite3@^7.6.13 \
@types/jsonwebtoken@^9.0.10 \
```

```
@types/leaflet@^1.9.20 \  
@types/multer@^2.0.0 \  
@types/nodemailer@^6.4.17 \  
bcryptjs@^3.0.2 \  
better-sqlite3@^12.2.0 \  
class-variance-authority@0.7.1 \  
clsx@2.1.1 \  
cmdk@1.1.1 \  
dotenv@16.4.7 \  
esbuild@0.25.1 \  
express@5.1.0 \  
express-rate-limit@^8.0.1 \  
express-validator@^7.2.1 \  
framer-motion@^12.23.6 \  
helmet@^8.1.0 \  
jsonwebtoken@^9.0.2 \  
kysely@^0.28.2 \  
leaflet@^1.9.4 \  
lucide-react@0.474.0 \  
multer@^2.0.1 \  
nodemailer@^7.0.5 \  
react@18.2.0 \  
react-day-picker@8.10.1 \  
react-dom@18.2.0 \  
react-router-dom@^7.6.3 \  
socket.io@^4.8.1 \  
socket.io-client@^4.8.1 \  
tailwind-merge@3.2.0 \  
tailwindcss-animate@1.0.7
```

This will add your dependencies to `package.json` and create/update `package-lock.json`.

B. Editing `package.json` Directly

Alternatively, add entries manually under `dependencies` and `devDependencies` (distinguish type declarations and build tools).

Example:

```
"dependencies": {  
  "@googlemaps/js-api-loader": "^1.16.10",  
  "@radix-ui/react-avatar": "^1.1.10",  
  "bcryptjs": "^3.0.2",  
  "better-sqlite3": "^12.2.0",  
  ...  
},  
"devDependencies": {  
  "@types/bcryptjs": "^2.4.6",  
  "@types/better-sqlite3": "^7.6.13",  
  ...  
}
```

After editing, run:

```
npm install
```

to ensure local files are added to `node_modules`.

4. Commit and Push to GitHub

1. Check changes:

```
git status
```

2. Add and commit updated lockfiles:

```
git add package.json package-lock.json
git commit -m "Add project dependencies"
git push
```

5. Best Practices

- Always commit both `package.json` and `package-lock.json` for full reproducibility.
- Avoid committing your modules folder.
- You can enable GitHub's Dependabot to keep dependencies updated and secure.

6. Summary Table (Install Examples)

Command Type	What To Do
Install single dep.	<code>npm install <package>@<version></code>
Install all at once	<code>npm install ...</code>
After manual edit	<code>npm install</code>

By following these steps, your GitHub project will be fully synced with your required dependencies, enabling hassle-free setup for collaborators and production environments.

: <https://docs.github.com/en/code-security/supply-chain-security/keeping-your-dependencies-updated-automatically>

