MIRACLE
SOFTWARE SYSTEMS

Simplifying Authentication by Integrating Google SSO in React

Hands-on Lab | Digital Summit 2024

Goal

Integrating Google SSO in React simplifies authentication by allowing users to log in securely with their Google accounts, enhancing convenience and security. It leverages Google's OAuth 2.0 for token-based authentication, reducing backend workload and development time. This approach provides a scalable, user-friendly experience while ensuring robust security. It eliminates the need for managing user credentials, streamlining the login process.

Pre-Requisites

The following installations will needed to complete this lab and run successfully,

- Google Account
- Node JS and NPM installed
- React JS
- Any Text Editor(VS Code/Notepad++/Notepad)

Technology Involved

- HTML
- CSS
- JavaScript
- React 18

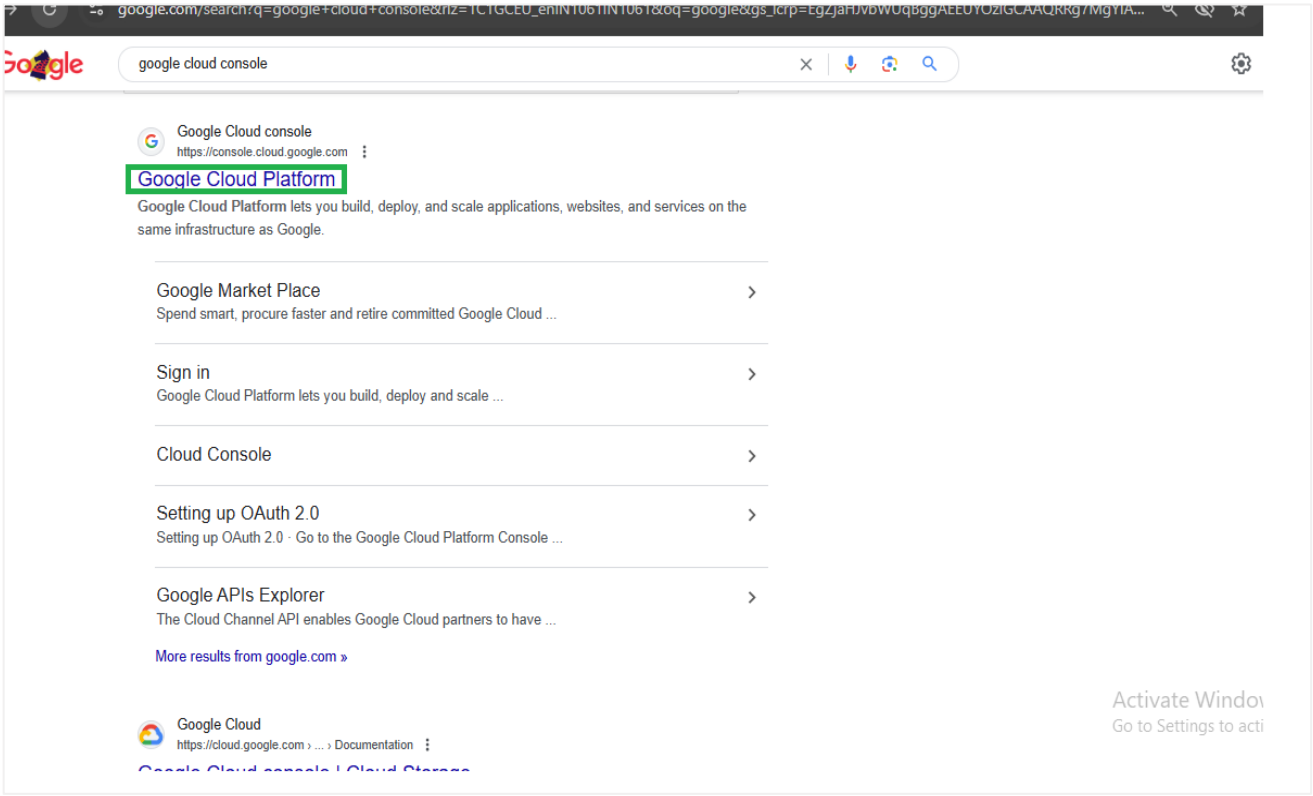
Lab Steps

Let's get started with the lab!

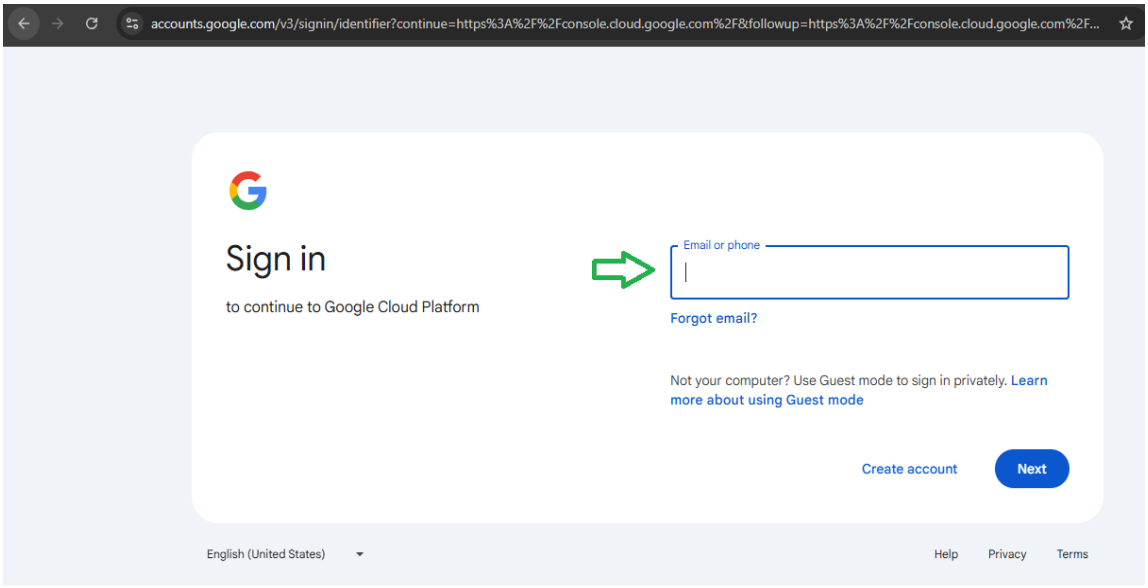
Step #1 | Set up Google Project

To set up Google projects in the Google Cloud Console, you need a Google account. Create an account if you don't have one or sign in with your existing Google Account.

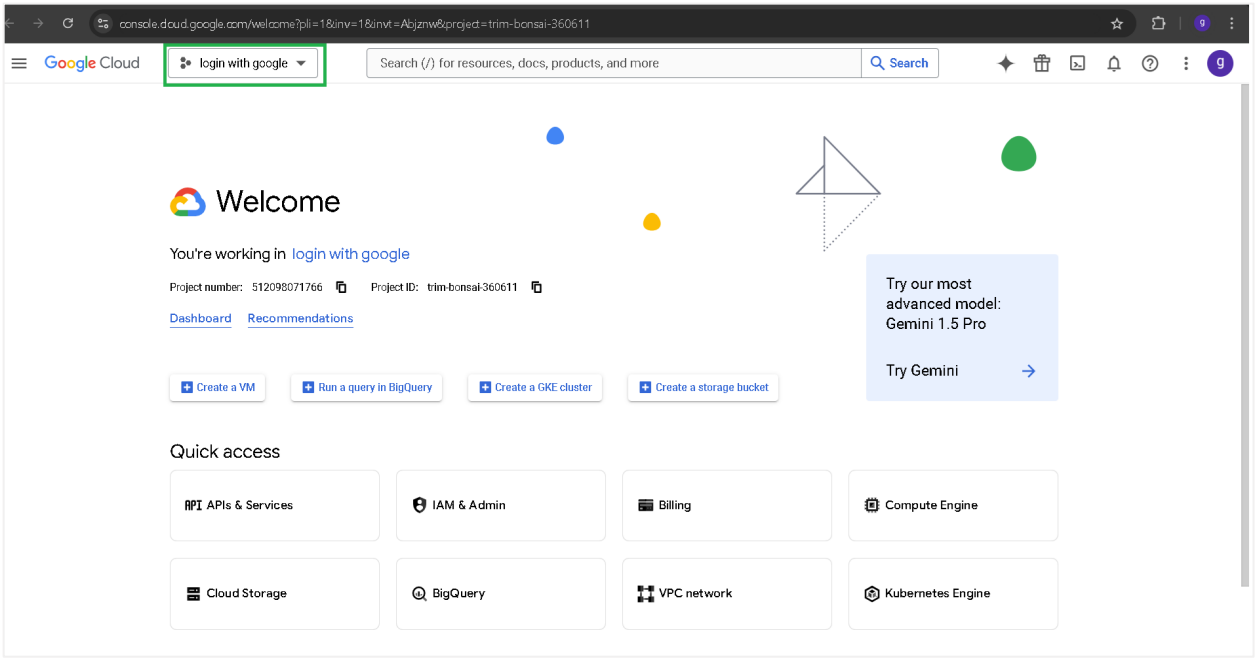
Go to Google Cloud Console.



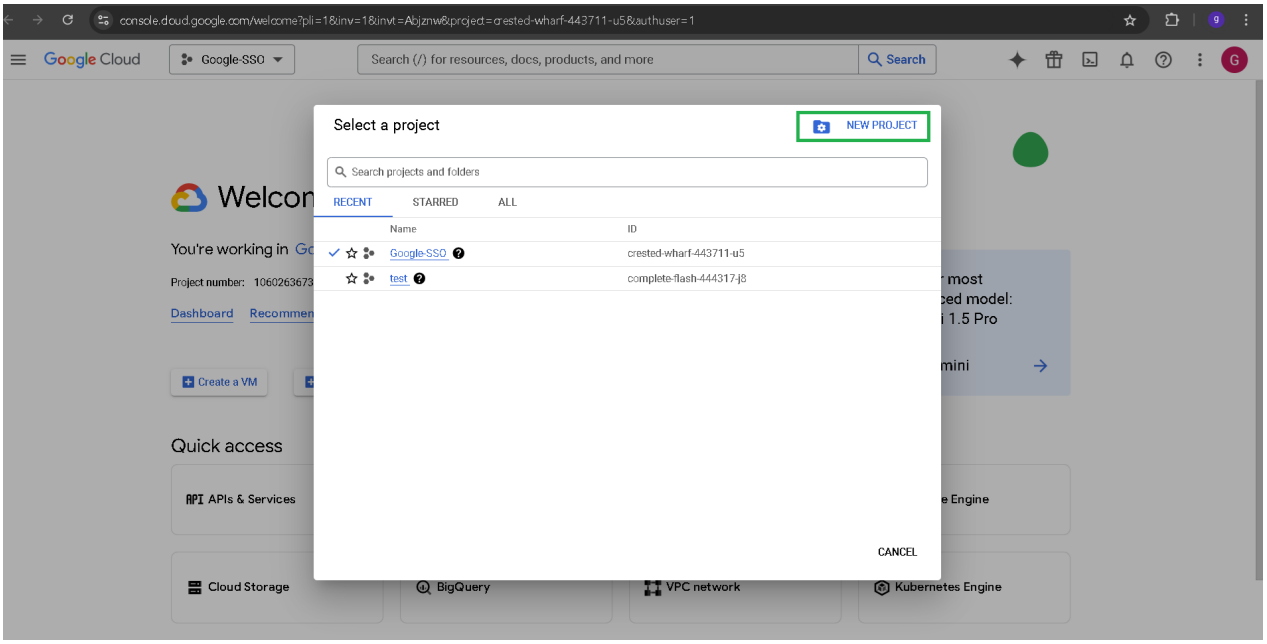
Sign in with your **Google Account**.



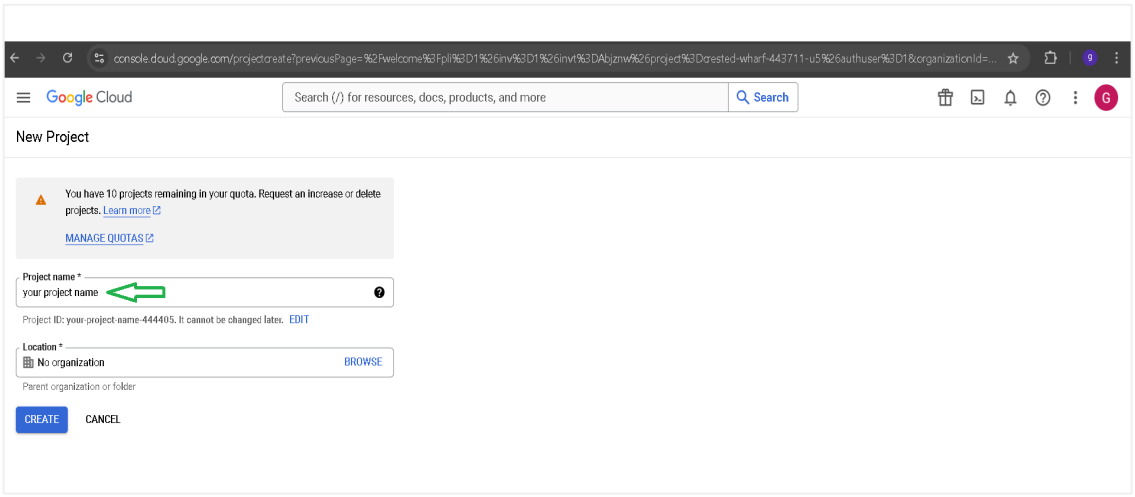
Create a **New Project**. In the top navigation bar, click on the **Project dropdown**.



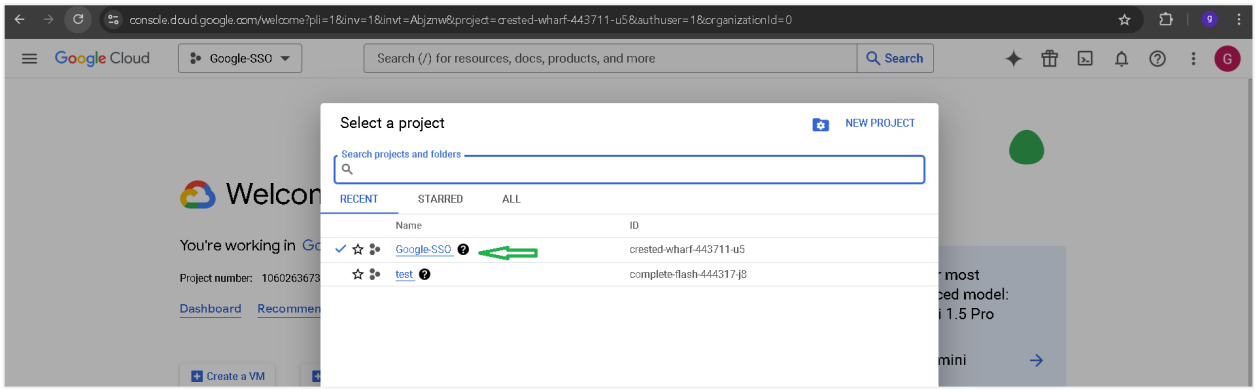
Click on **New Project.**



Enter a **Project Name** and click on **Create** to create a new project.

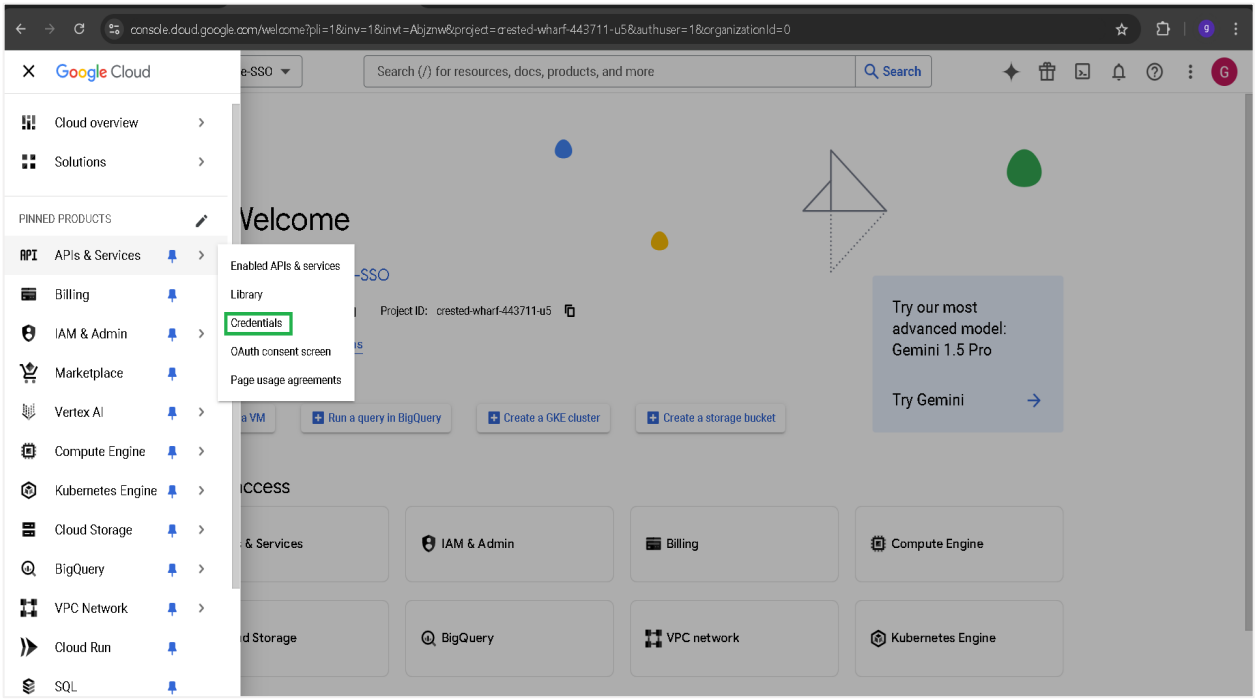


Here, we can see that the project has been created successfully.

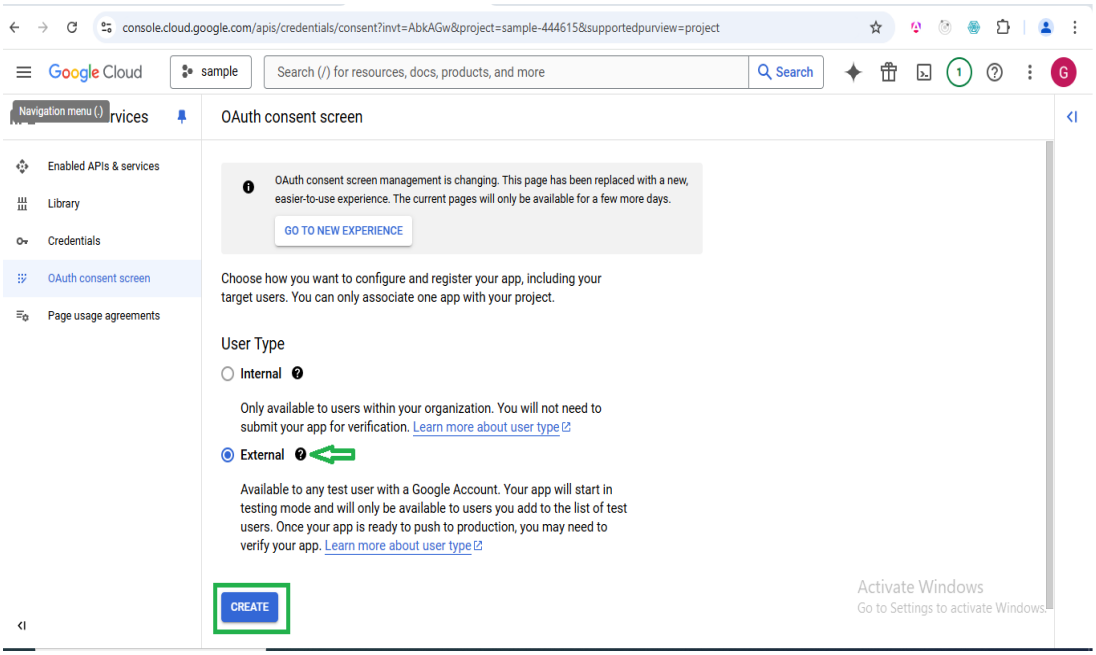


Step #2 | Configure OAuth Consent Screen

In the Google Cloud Console, click on the **Navigation Menu** in the top left corner. Go to **APIs & Services -> OAuth consent screen**.

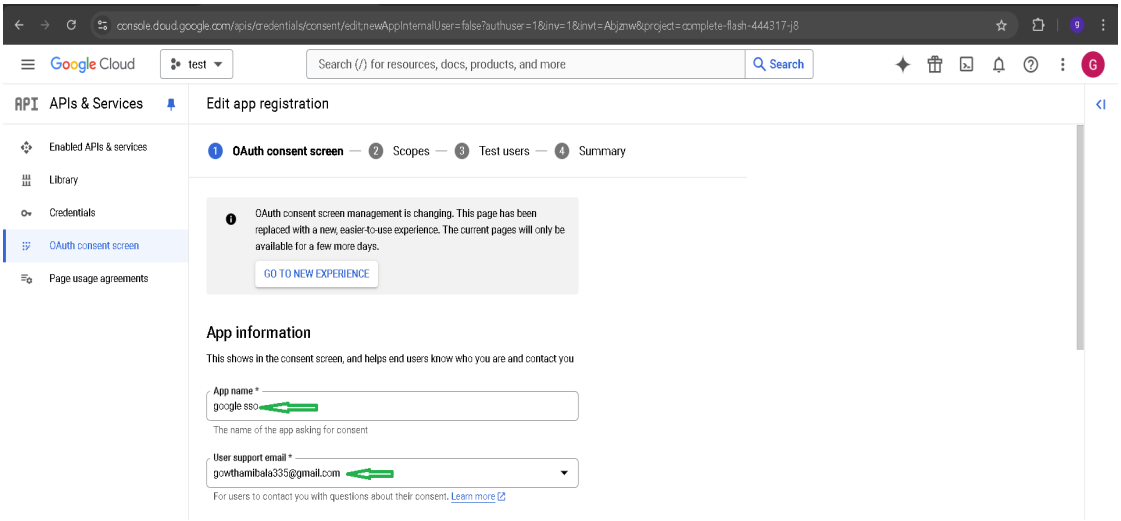


Here, we have to set up the OAuth consent screen. Select the **External** option if your app is for external users (outside your organization). Otherwise, select **Internal** if it's for your organization. Click on **Create**.



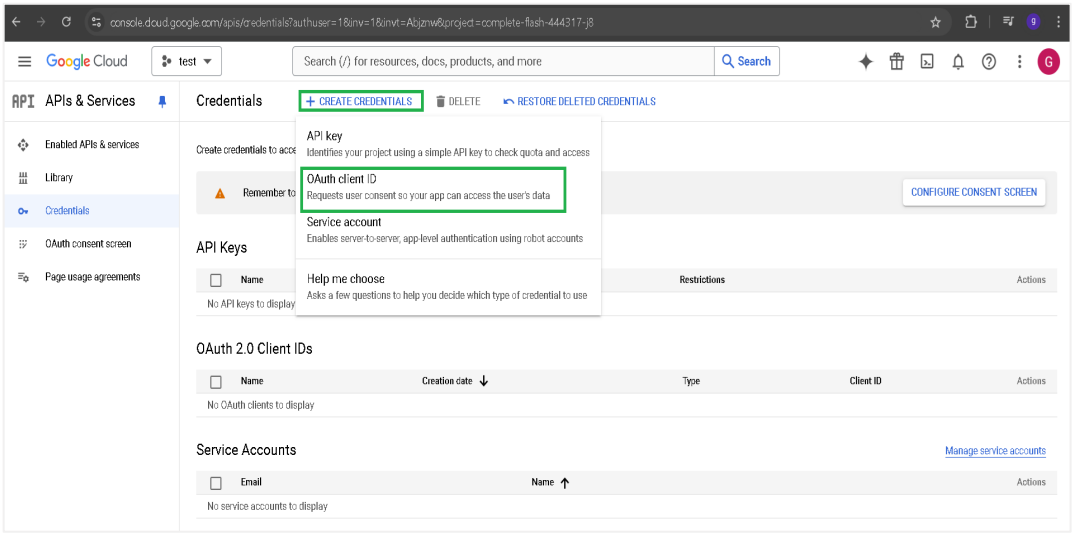
Fill in the required information,

- **App name -** The name of your app
- **User support email -** Email address for user support
- **App logo (optional) -** Upload your app's logo
- **Scopes -** Add any necessary Google API scopes (e.g., email, profile)
- **Test users -** If your app is in testing mode, add email addresses of users who will be allowed to log in during this phase

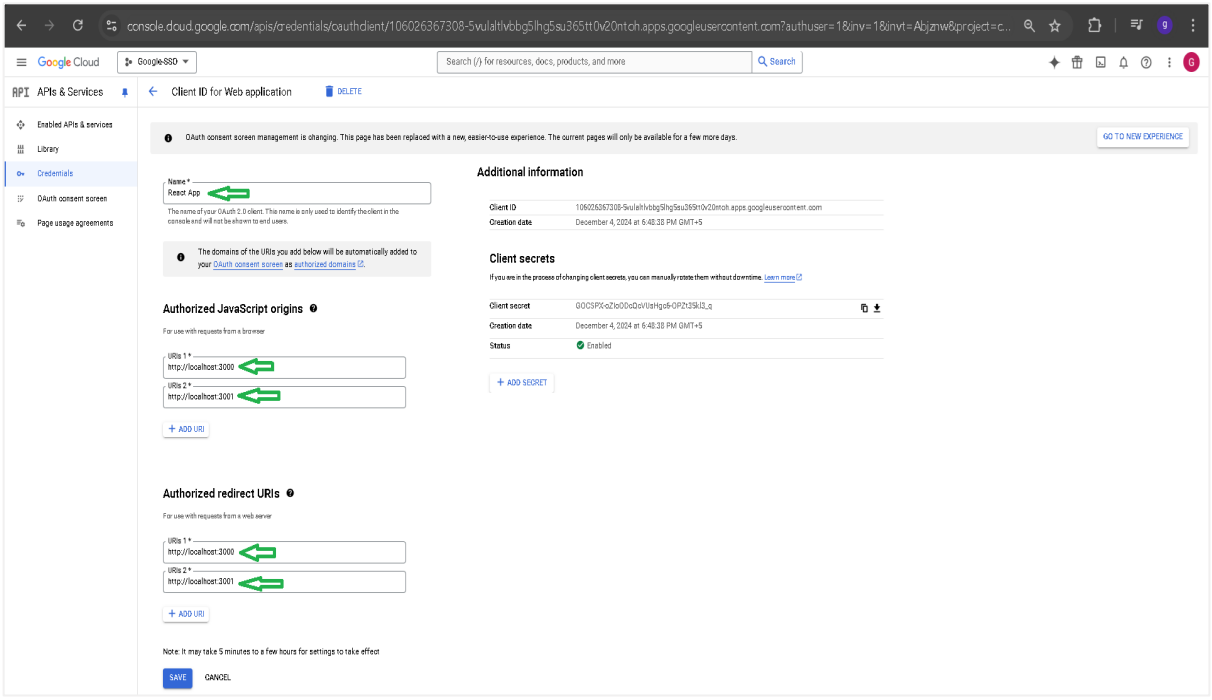


Step #3 | Configure OAuth 2.0 Client ID

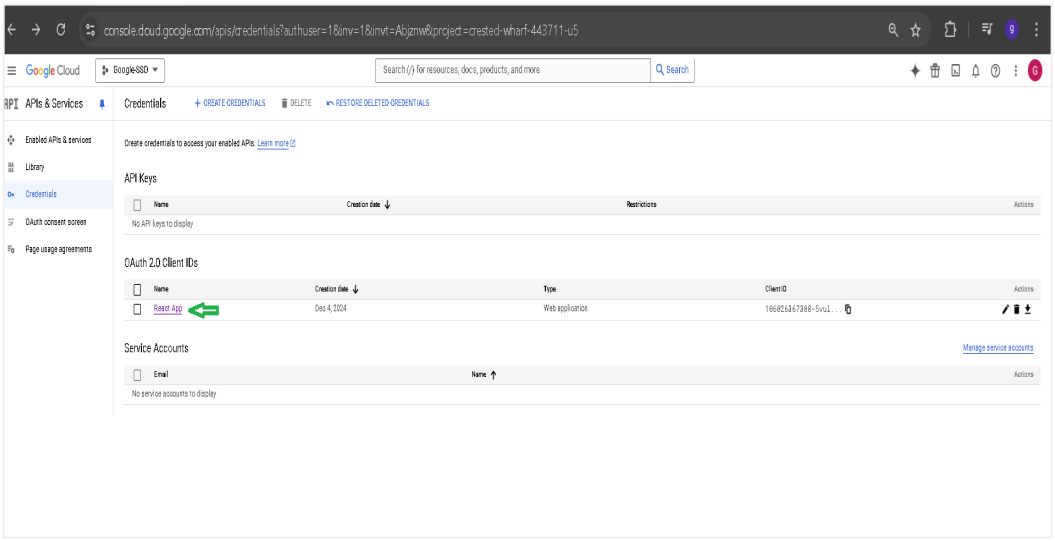
Click on **Credentials** and then click on **CREATE CREDENTIALS.**



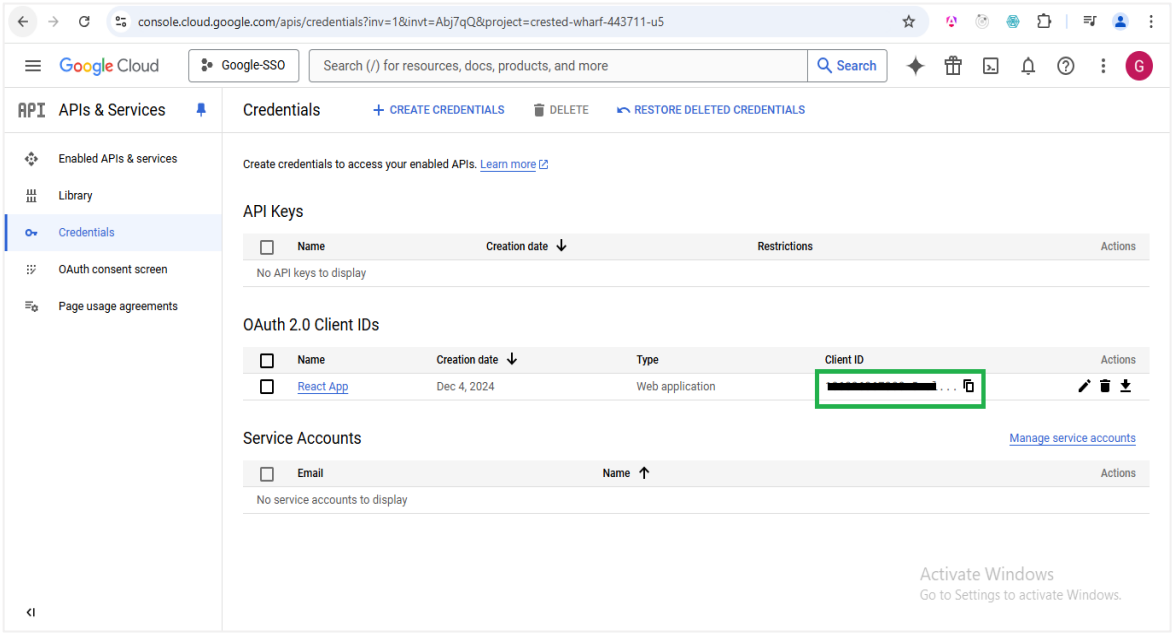
Select OAuth client ID.



Choose Web Application as the application type. Add your React app's local URLs **(http://localhost:3000 & http://localhost:3001)** under Authorized redirect URIs. Click on **CREATE**.



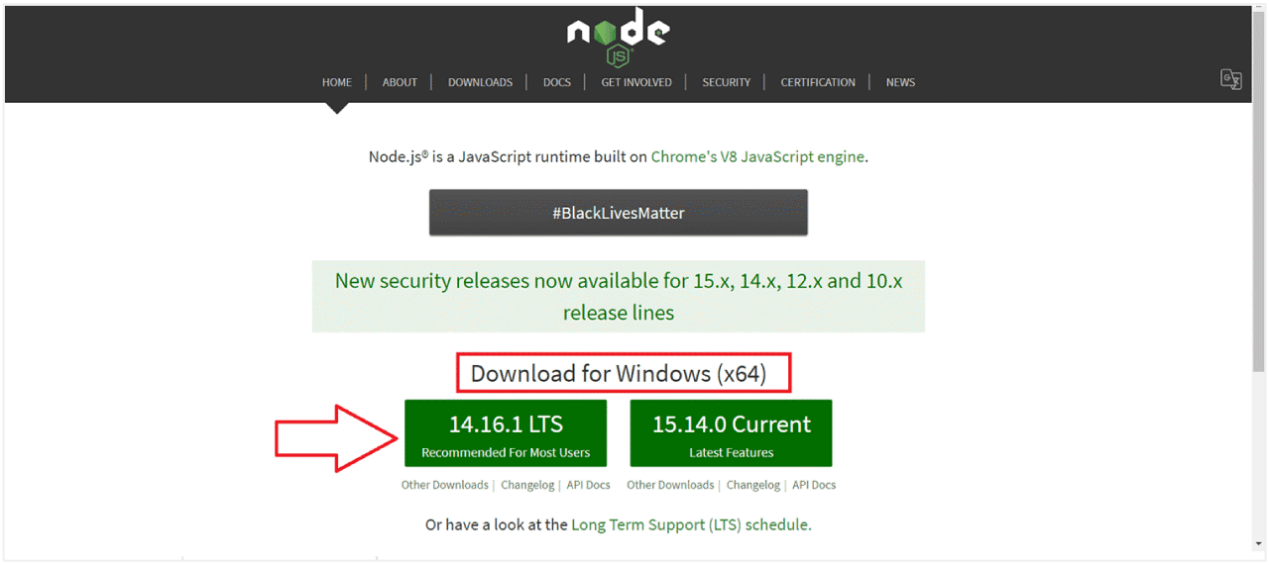
Now, OAuth Client ID has been created.

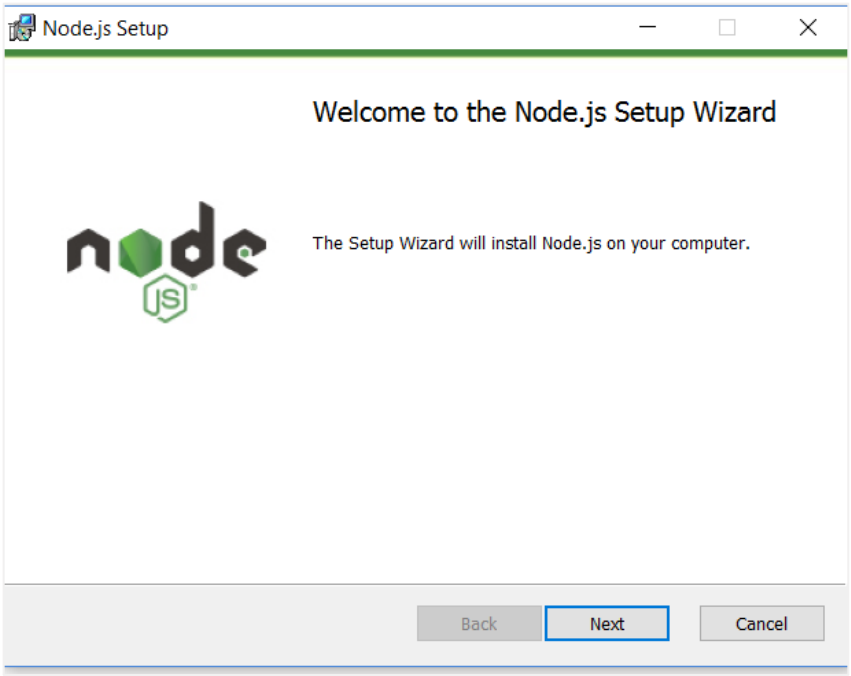


Save your **Client ID**.

Step #4 |Install Node.js and NPM

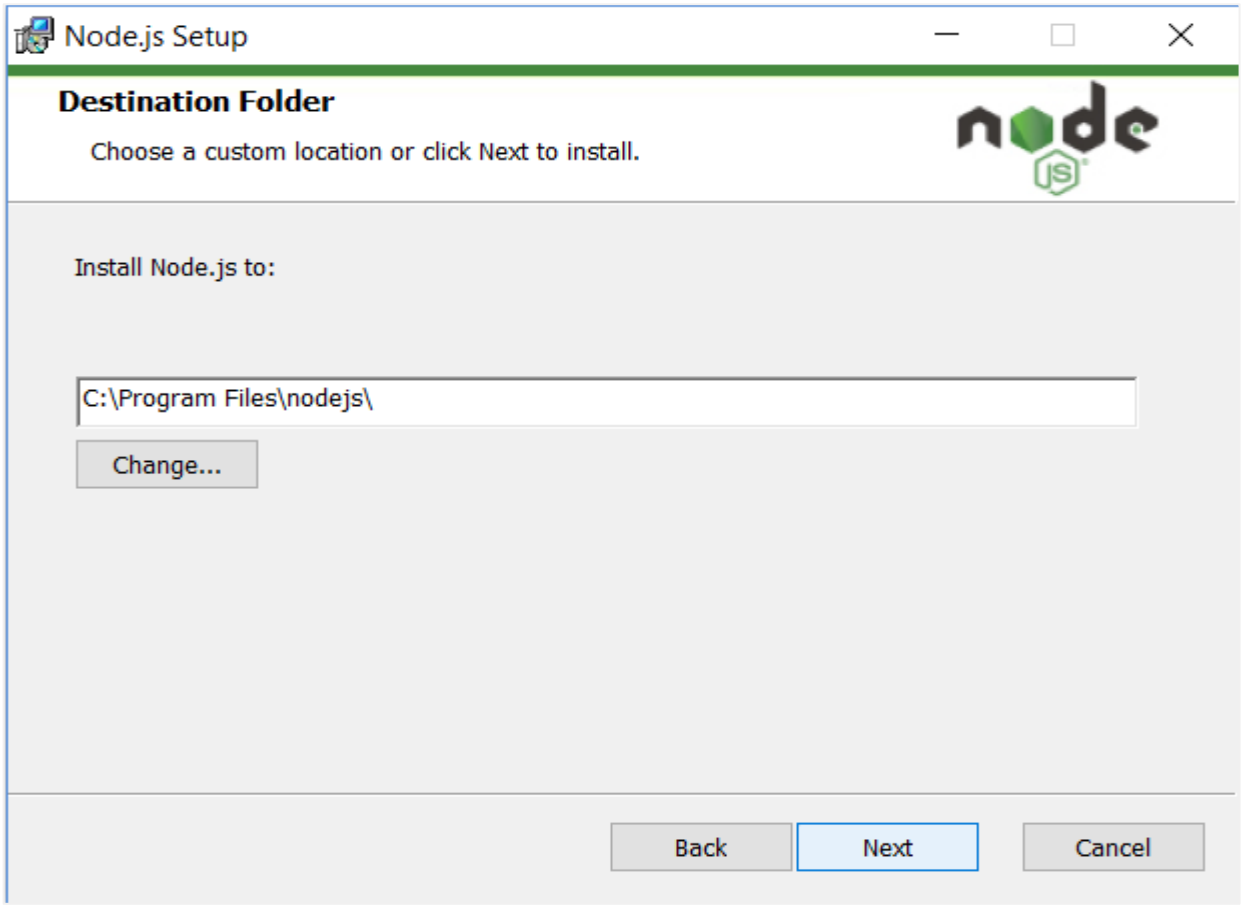
- Go to the official Node.js website: https://nodejs.org/en
- Download the latest version of Node.js for Windows. Make sure you select the 64-bit version
- This installer also includes NPM, which is the package manager used with Node.js





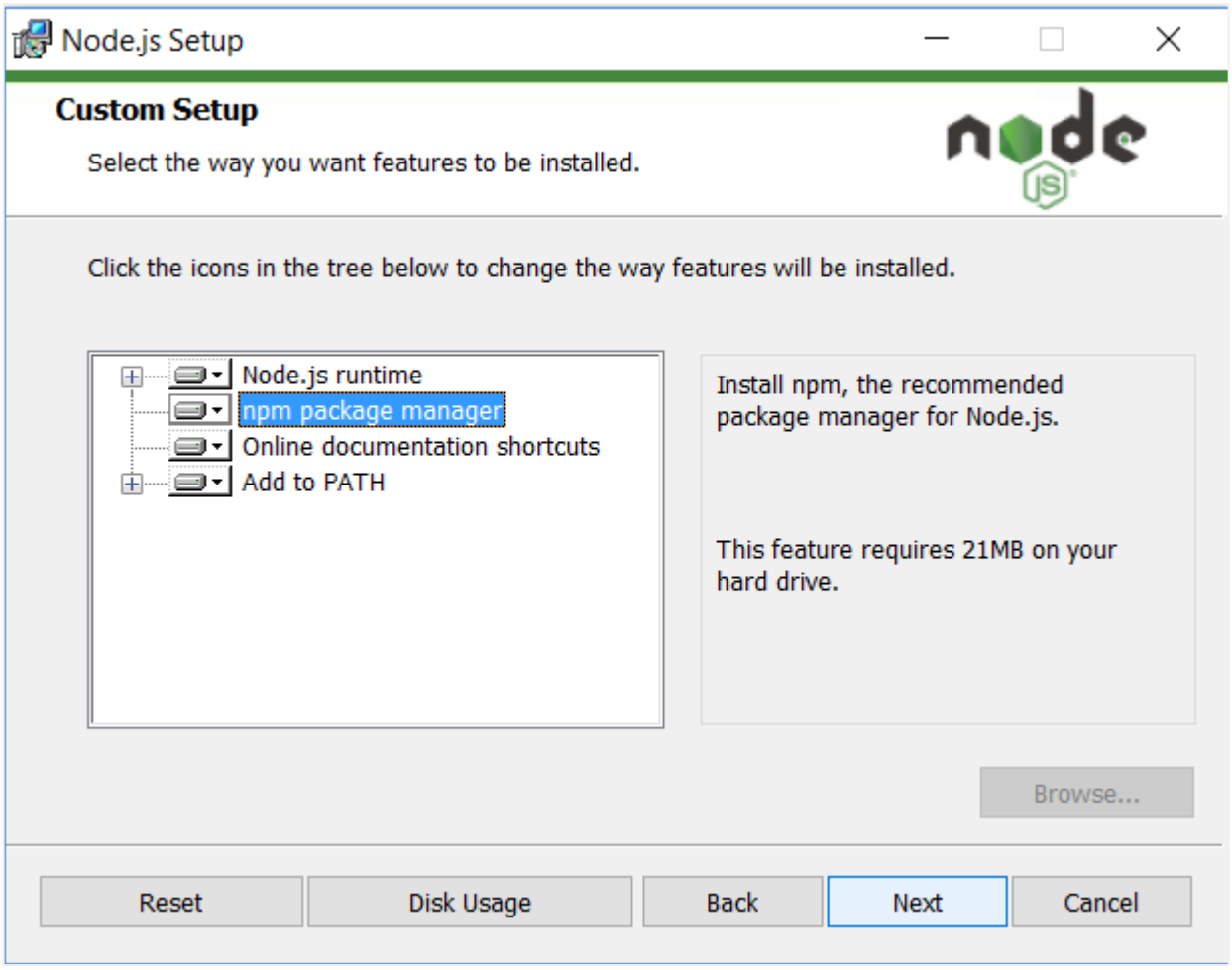
- After choosing the installation path, double-click the .msi file to start the installation
- Allow the application to run
- A welcome message will appear on your screen. Click the "Next" button to begin the installation process

Choose the desired path where you want to install Node.js.

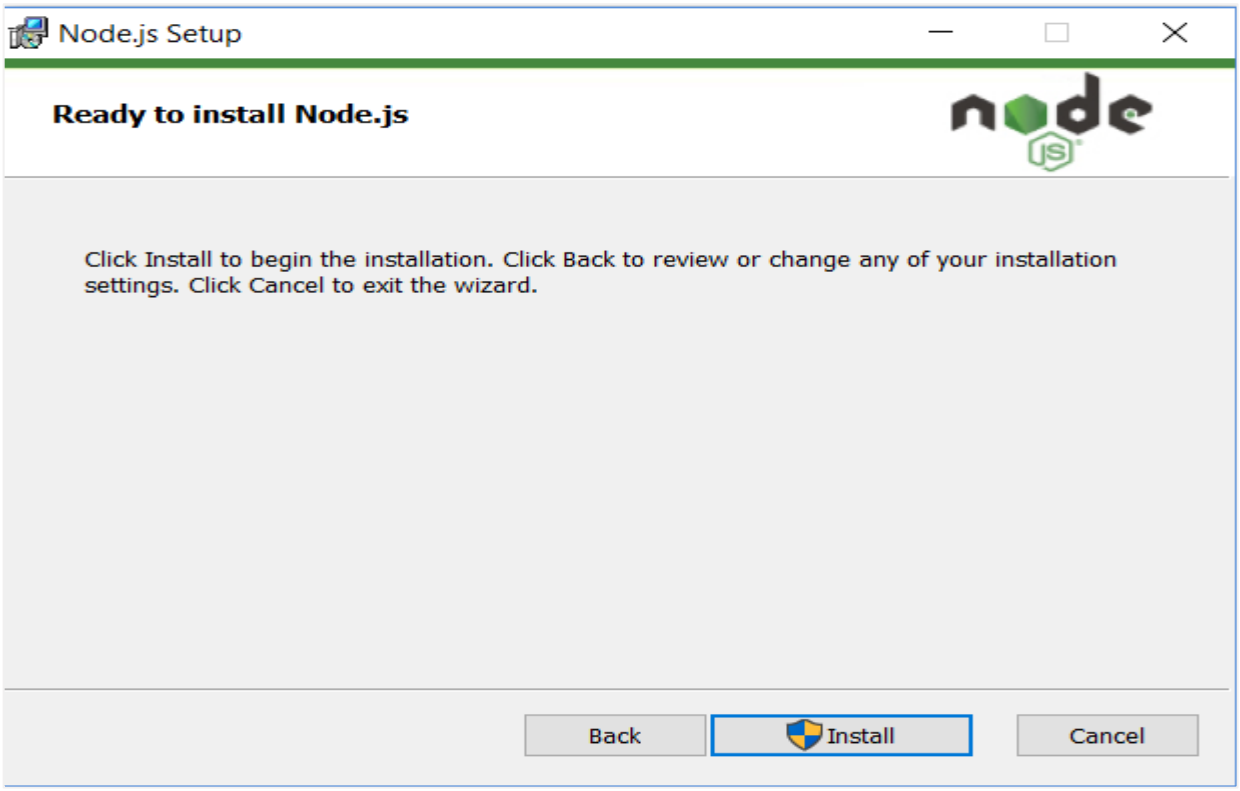


- Click the "Next" button to see the custom setup page
- Select the npm package manager option instead of the default Node.js runtime
- This will let you install Node.js and NPM together,
- The following features will be installed by default,
  - Node.js runtime
  - Npm package manager

- Online documentation shortcuts
- Add to Path

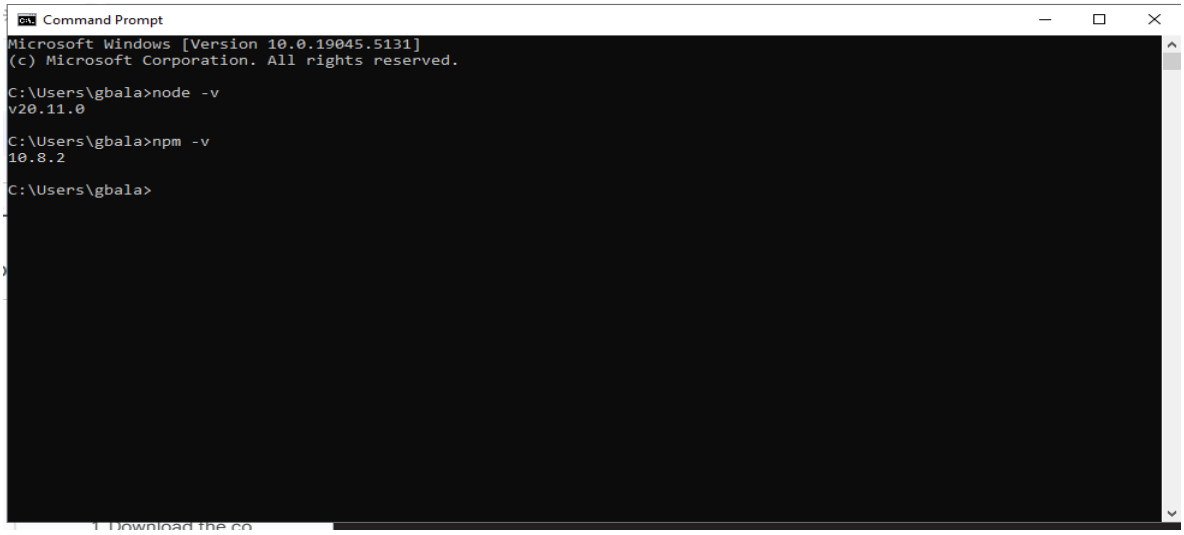


The setup is ready to install Node and NPM. Let's click on the Install button so hard!



Step #5 | Check Node JS and NPM Version

To open the command prompt (Terminal), type "cmd" in the search bar, and you will find the command prompt.



- Run the following commands to verify installation,

- node -v -this shows the installed version of Node.js
- npm -v - this shows the installed version of npm

Step #6 | Code Setup in Local Machine

1. Download the code from GitHub

```
// GitHub repository URL
// Download as Zip screenshot
// Unzip code
```

2. Set Up Project Structure

Create the following files in the existing code,

`.env`

3. Environment Variables (.env)

- Add **REACT_APP_GOOGLE_CLIENT_ID** as env variable
- Copy the **Client ID** from **Google Cloud Console** that you have created
- Place **your_client_Id** in **.env** file

```
REACT_APP_GOOGLE_CLIENT_ID='your_client_id'
```

4. Install Modules and Run the React App

```
npm install
```

```
PS C:\Users\gbala\myprojects\summit\final\Google-SSO> npm install

up to date, audited 1425 packages in 8s

274 packages are looking for funding
  run `npm fund` for details

10 vulnerabilities (4 moderate, 6 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force
```

```
npm start
```

```
C:\Users\gbala\myprojects\react\google_sso> npm start
```

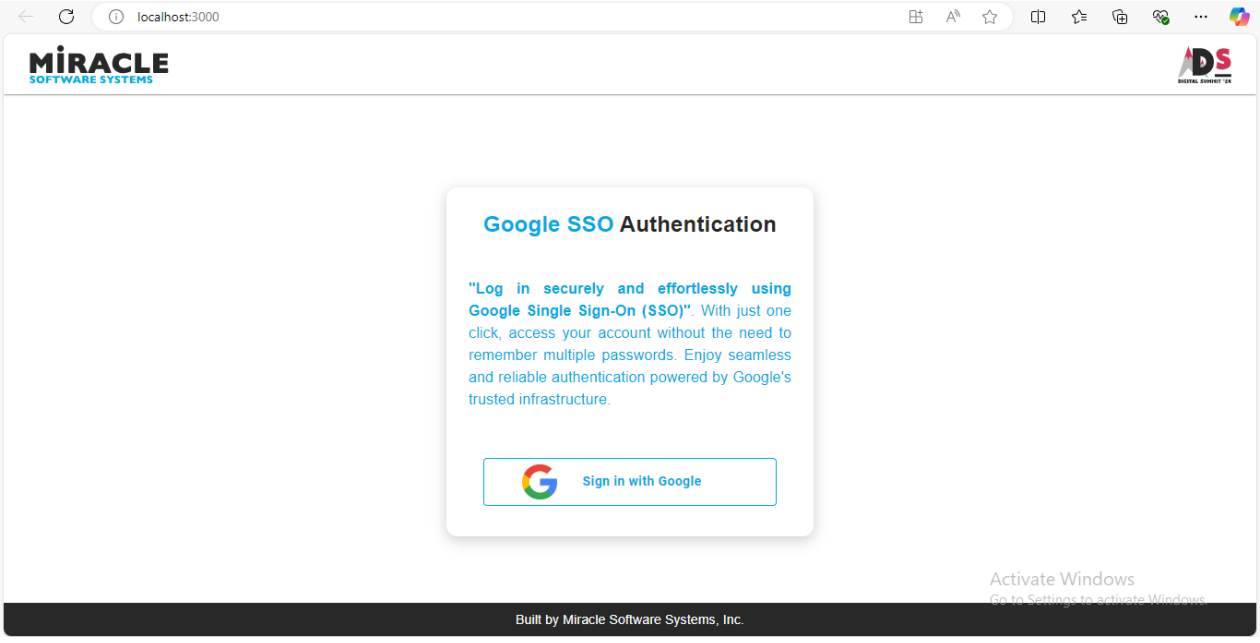
```
Starting the development server...
Compiled successfully!

You can now view google_sso in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://172.17.2.37:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

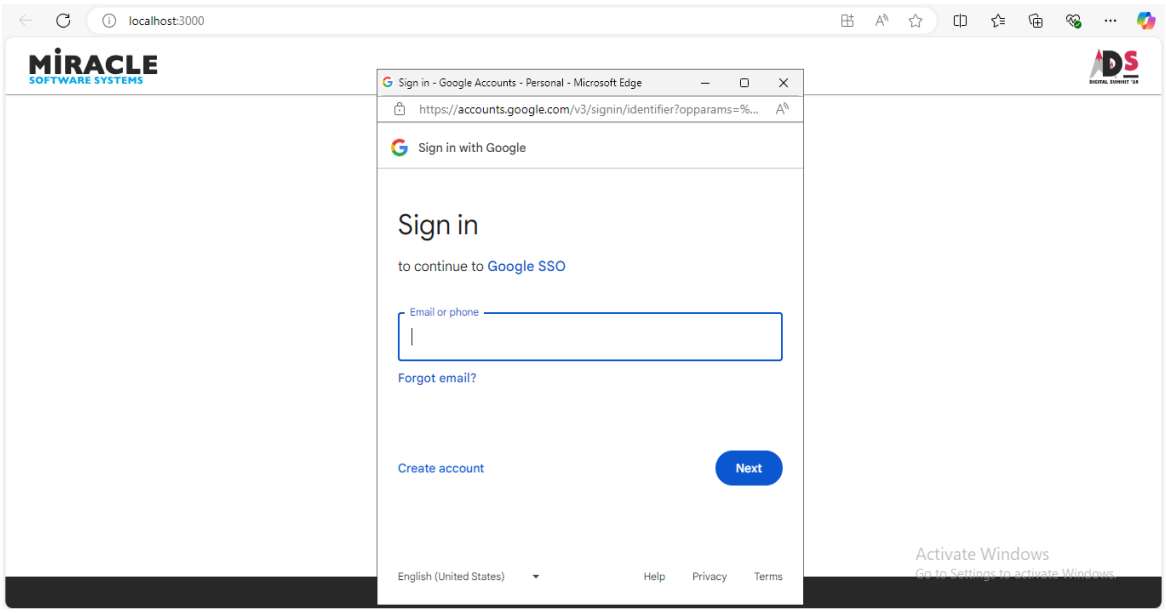
5. Please follow the same steps for Application 2 as you did for Application 1.

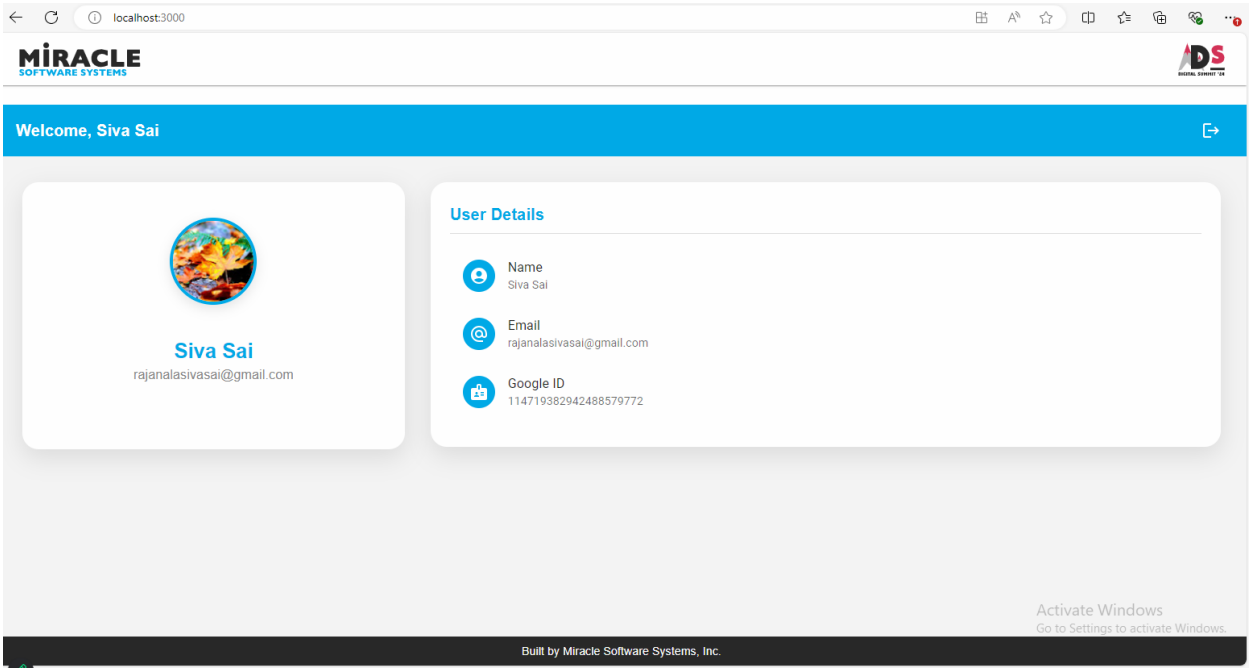
6. Run and test the React Application1 -

Run this command for First Application: **npm start - -port 3000**
Open the frontend in your browser at **http://localhost:3000**

Step #7 | Screens of Application #1





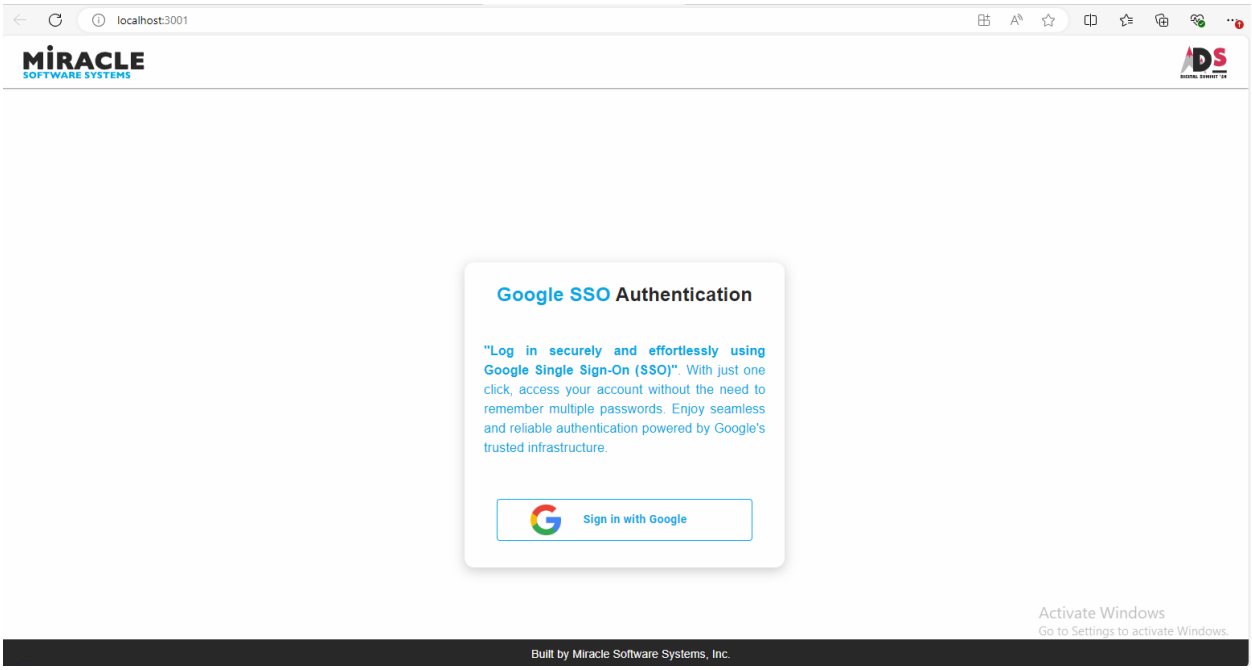


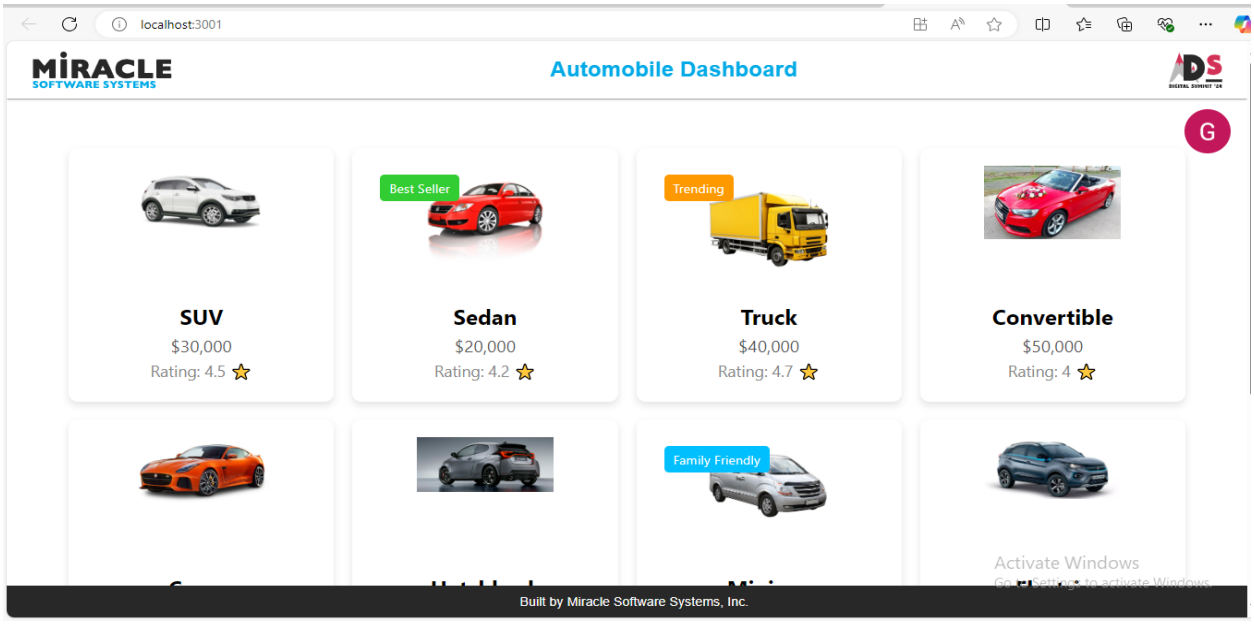
7. Run and test the React Application2

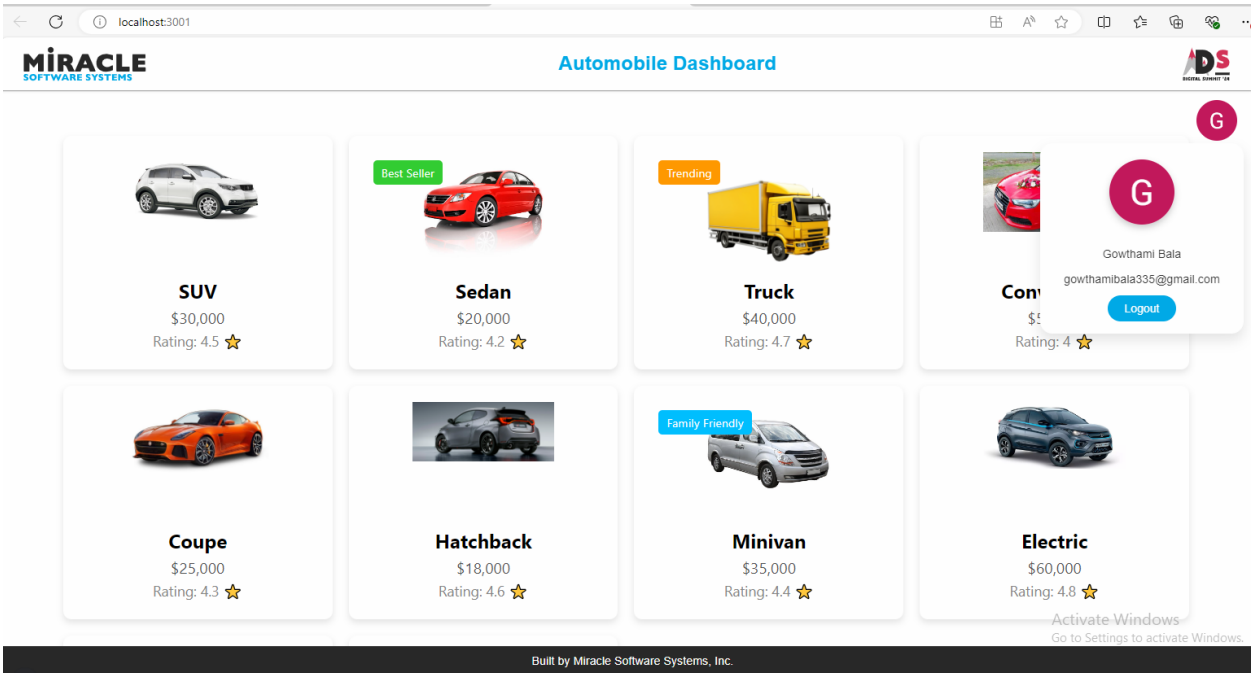
Run this command for Second Application: **npm start - -port 3001**

Open the frontend in your browser at http://localhost:3001.

Step #8 | Screens of Application #2






localhost:3001
MIRACLE
SOFTWARE SYSTEMS
Automobile Dashboard
G
Gowthami Bala
gowthamibala335@gmail.com
Logout
Best Seller
Trending
SUV
$30,000
Rating: 4.5
Sedan
$20,000
Rating: 4.2
Truck
$40,000
Rating: 4.7
Rating: 4
Family Friendly
Coupe
$25,000
Rating: 4.3
Hatchback
$18,000
Rating: 4.6
Minivan
$35,000
Rating: 4.4
Electric
$60,000
Rating: 4.8
Activate Windows
Go to Settings to activate Windows.
Built by Miracle Software Systems, Inc.