A black background with blue and red text

Description automatically generated

College of Computing and Data Science

Nanyang Technological University

**Event calendar builder**

**(README FILE)**

**CCDS24-0288**

Ng Wee Kiat

U2122504J

A Final Year Report submitted to the College of Computing and Data Science, Nanyang Technological University in partial fulfilment of the requirements for the Degree of

*DEGREE OF BACHELOR OF COMPUTING (COMPUTER SCIENCE)*

Supervisor: Dr Shen Zhiqi  
Examiner: Asst Prof Wei Dong

Demo Video: <https://youtu.be/V2nJkcDVbOY>

# Acknowledgement

I would like to express my deepest gratitude to my supervisor, Dr Shen Zhiqi, for his invaluable guidance, support, and encouragement throughout my final year project. Dr Shen Zhiqi has been instrumental in shaping this project, offering insightful advice and constructive feedback that allowed me to navigate the challenges of designing and developing the event calendar builder. This project, which explores the intersection of text mining, natural language processing, and calendar integrations, has been both a challenging and rewarding experience. Dr Shen Zhiqi not only helped refine my technical skills but also inspired a deeper passion for developing practical applications that bridge technology with real-world needs.

I would also like to extend my gratitude to my peers and family for their unwavering support, as well as to the faculty and staff of Dr Shen Zhiqi, who provided the resources and knowledge base to make this project possible. Finally, I am grateful for this opportunity to apply and expand my skills, contributing to a tool that simplifies the management of events and schedules. This journey has been an enriching experience, and I am excited to continue exploring the potential of technology in the future.

Contents

[Acknowledgement 2](#_Toc196484501)

[Project Structure 4](#_Toc196484502)

[Web Application Setup 4](#_Toc196484503)

[Software Prerequisites 4](#_Toc196484504)

[Credential Prerequisites 5](#_Toc196484505)

[Running the application: 8](#_Toc196484506)

[Android Application Setup 9](#_Toc196484507)

[Software Prerequisites 9](#_Toc196484508)

[Credential Prerequisites 9](#_Toc196484509)

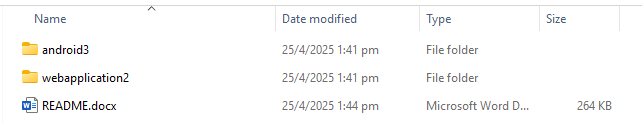
[Running the application: 11](#_Toc196484510)

[Final Notes 11](#_Toc196484511)

# Project Structure

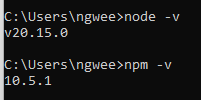
After downloading the project, extract the FYP\_Final package. It contains the following directories:

* Android3/ – Android application source code
* Webapplication2/ – Web application source code



# Web Application Setup

## Software Prerequisites

1. Install node.js (<https://nodejs.org/en>) on your **system**.   
   During installation, ensure the option “Add to PATH” is selected.  
   To Verify installation:  
   Open terminal/command prompt  
   **node -v  
   npm -v**
2. Install Node.js Extension in **VSCode**.

* Go to extensions tab
* Search for Node.js Extension Pack
* Install Node.js Extension Pack, ESLint, Npm Intellisense

1. Install dependencies using  
   Open your terminal, navigate to the Webapplication2/ directory, and run:  
   | **npm install**(Note: This may take 5+ minutes depending on your internet speed.

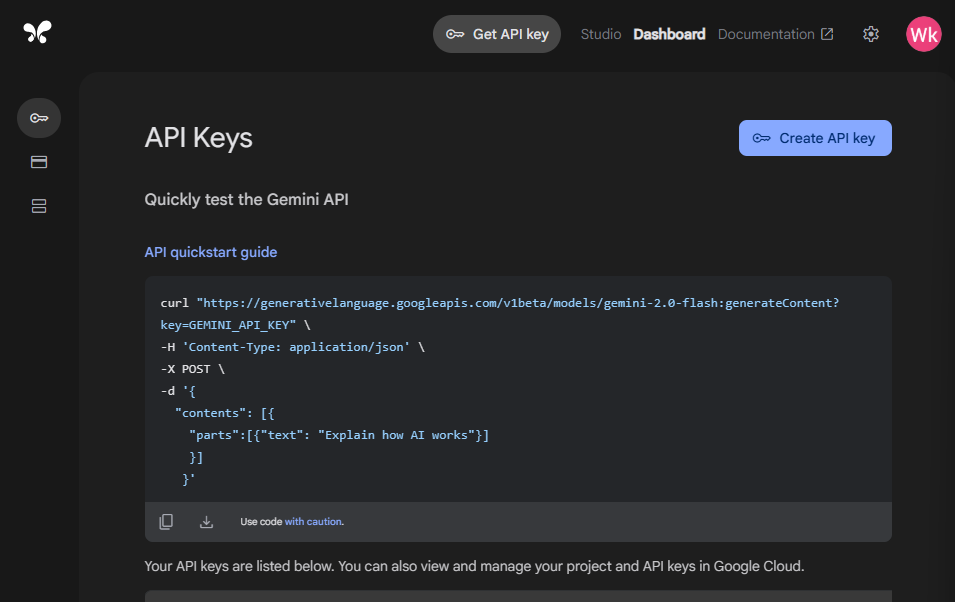
## Credential Prerequisites

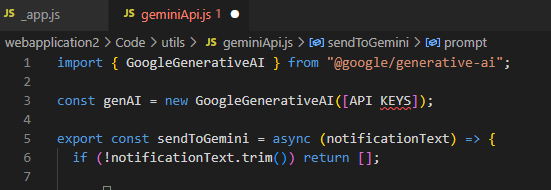
3 Credentials must be customed to the users profile to access google services

1. Google Gemini API Keys
2. Google cloud console ClientID
3. Google cloud console redirect link

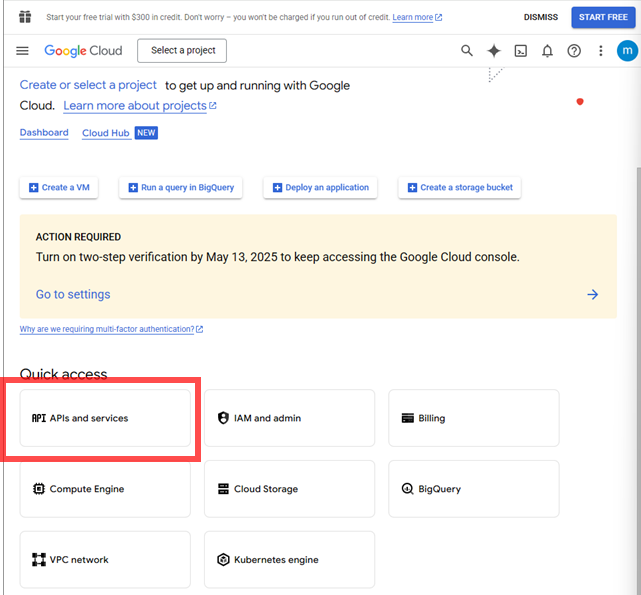
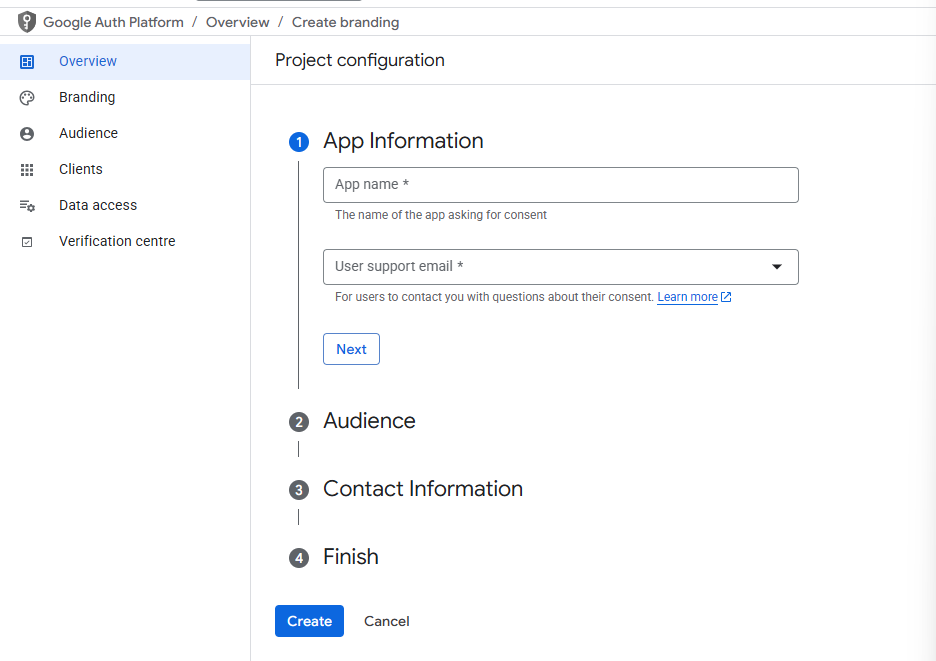
Google Gemini API Keys:

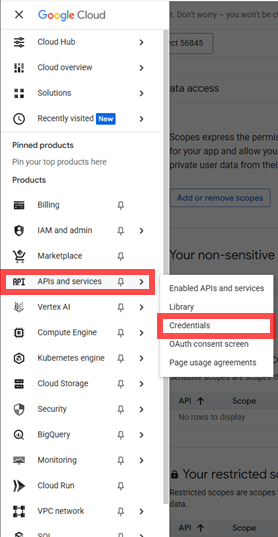
1. Visit Google Ai Studio (<https://aistudio.google.com/app/apikey>)
2. Click Create API Key



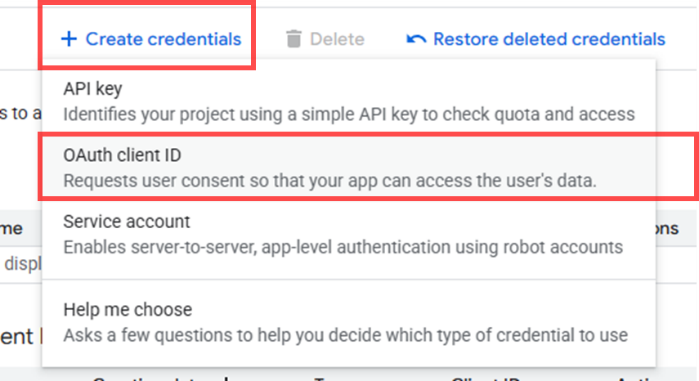
1. Select relevant project – If no existing project create a new project
2. Create API Key, copy API key to a notepad (Or any secure temporary location)
3. In the code files, access webapplication2 > Code > utils > geminiApi.js
4. Paste the Gemini API keys into the placeholder **[API KEYS]**  
   

To get the Google Cloud Console access:

1. Go to google cloud console (<https://console.cloud.google.com/>)
2. Under Quick Access -> APIs and Services  
   
3. Create Project  
   **Project Name**: Up to you
4. In the left column – Access OAuth consent screen  
   Click Get Started  
   Fill in relevant fields:  
   **App Name:** Up to you  
   **User Support Email:** Email your google cloud console is under  
   **Audience:** External (IMPORTANT)  
   **Contact Information Email:** Your google cloud console email  
   
5. Click the left panel -> APIs and Services -> Credentials



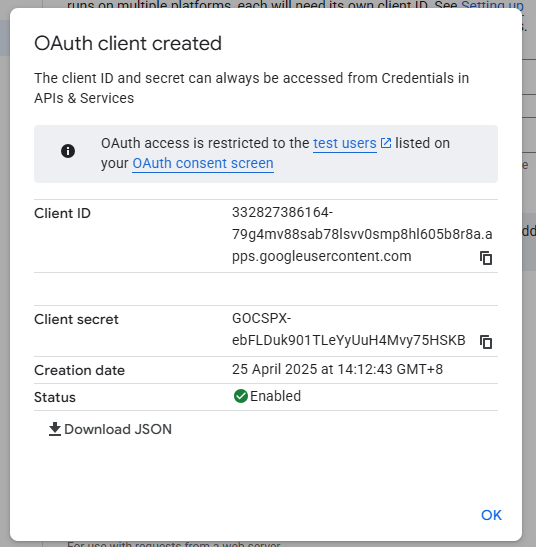
Click **“+Create Credentials”**

****

**Application type:** Web Application  
**Name:** Up to you  
**Authorized Javascript Origins(IMPORTANT):**http://localhost:3000http://localhost:3001 **Authorized redirect URLs(IMPORTANT):**http://localhost:3000  
http://localhost:3001

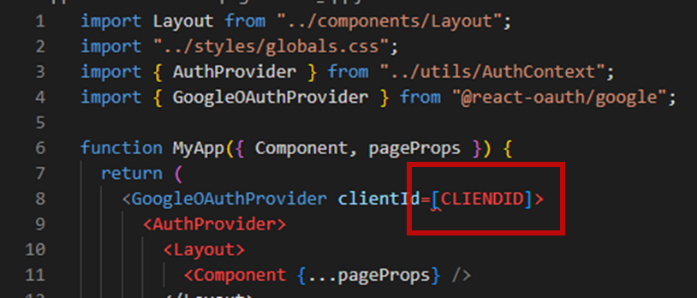
(Number of Local Host ports depends on whether you’re concurrently running multiple local host instances – Contingency set as 2 ports so if 3000 is being used, the app is still operational)

1. Upon creation you will receive the ClientID – Copy the client ID

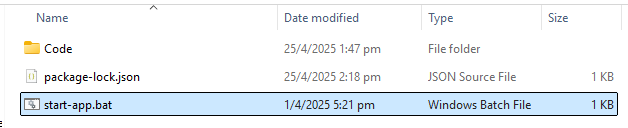


1. In the code base, open webapplication2 -> Code -> pages -> \_app.js

Paste the ClientID in the placeholder [CLIENTID]



## Running the application:

Run the **start-app.bat** file  


**OR**

Manually start the local host instance

1. Navigate your command prompt to the webapplication3 folder path
2. Run: **npm run dev**

For both methods the application will immediately open a new terminal with the node.js instance and the webpage will appear in a new Google Chrome/Default browser tab

# Android Application Setup

## Software Prerequisites

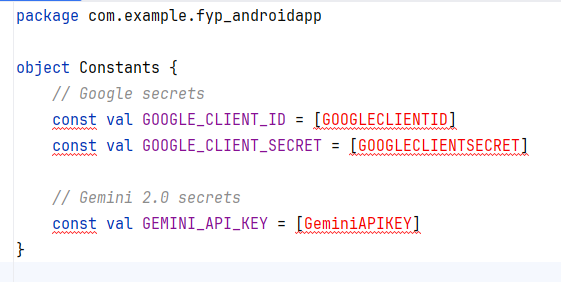
1. Install Android Studio (<https://developer.android.com/studio>)

(At time of development Android studio Meerkat was used)

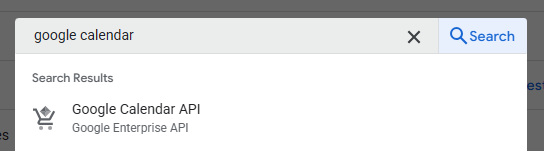
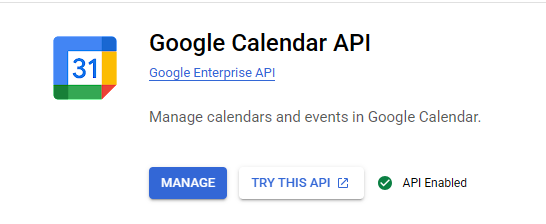
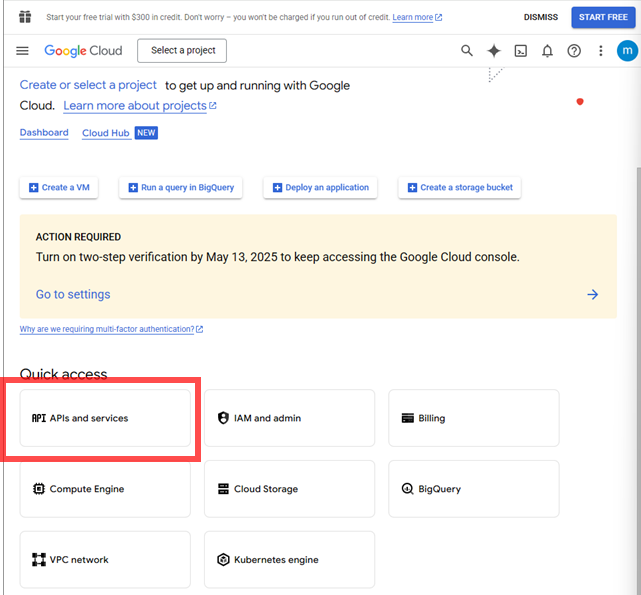
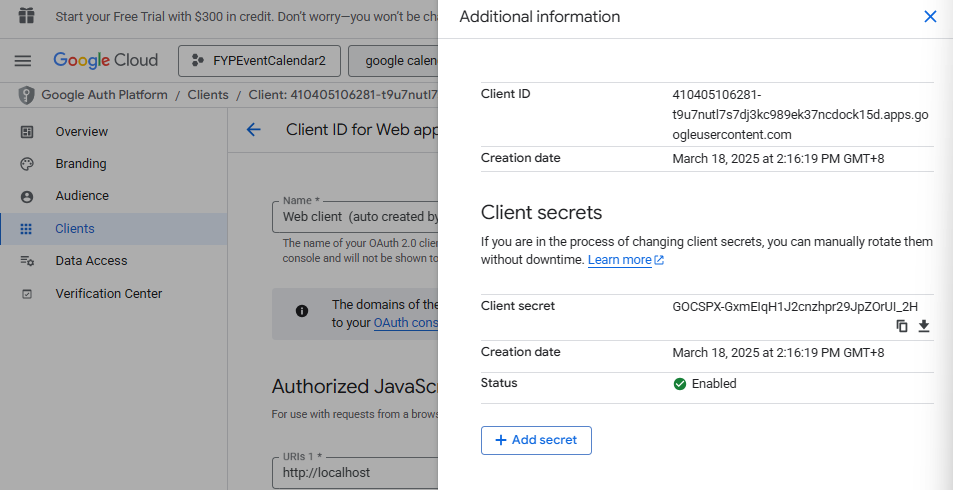
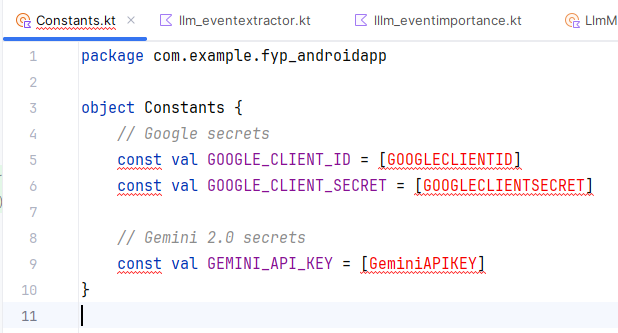
1. Run the setup, and allow the build gradle to run
2. Open the folder android3 in android studio
3. Initialization will take awhile for setup to be done

## Credential Prerequisites

To setup the Google Gemini API keys see above for retrieval of API keys

1. Setup API keys in app -> Constants.kt -> Paste into GeminiAPIKEY  
   

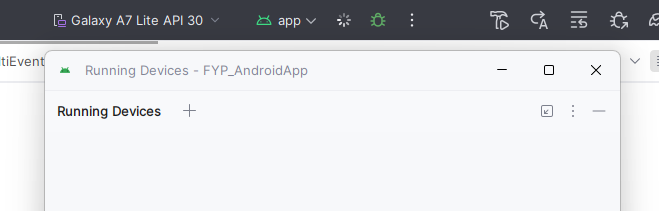
To setup Google Cloud Console APIs:

1. Go to google cloud console (<https://console.cloud.google.com/>)
2. In the search bar -> search: Google calendar API  
     
   Enable the Google calendar API  
   
3. Under Quick Access -> APIs and Services  
   
4. Access the same API credentials previously created
5. Click the (i) button on the upper right hand corner  
   Copy the Client ID and Client secret from the pop out bar  
   
6. Paste them into Constants.kt  
   

## Running the application:

1. Enable USB debugging on your android phone  
   (Note: Xiaomi, Hua Wei, and other phones running China’s App store will not work – Phone requires Google Play Store for the Application)  
   Google the steps – Varies too much phone by phone to include here
2. Plug phone into laptop

Wait for phone to be detected by android studio

1. Request will appear for data to be accessed by the laptop  
   (Authorize the connection)
2. Click the green run button  
   
3. Application will start on the android device
4. Unplug the cable

# Final Notes

* Ensure both your web and Android projects use **the same Google Cloud Console project** for consistent API access.
* If port 3000 is already in use, port 3001 will serve as a fallback.
* For any CORS or auth errors, verify that:
* Client IDs are correctly inserted
* Redirect URIs and origins match in the Google Cloud Console