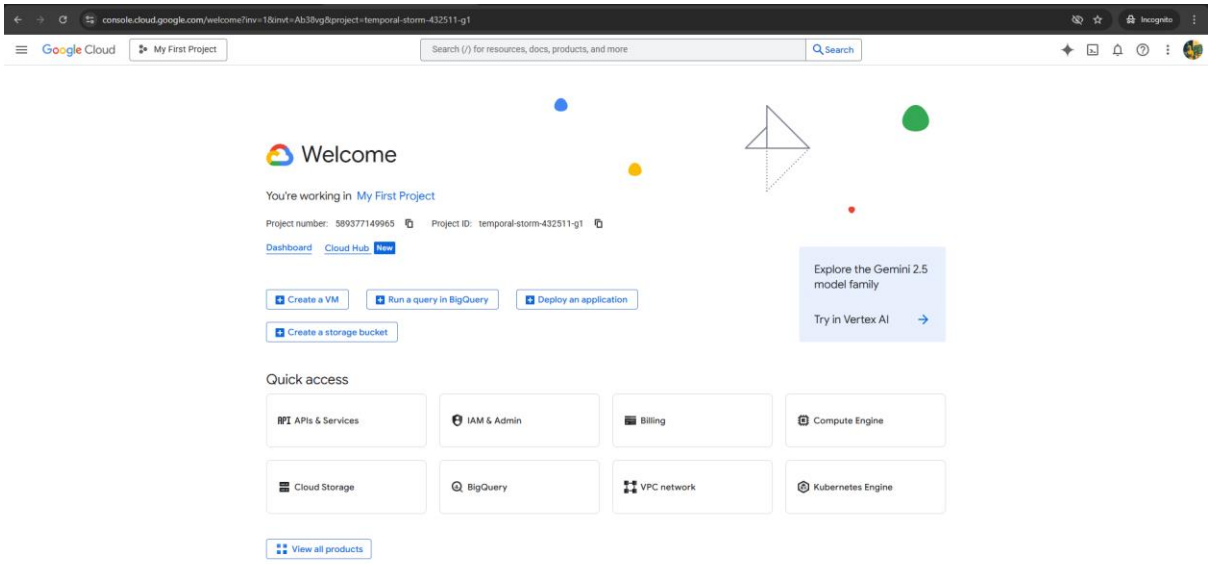
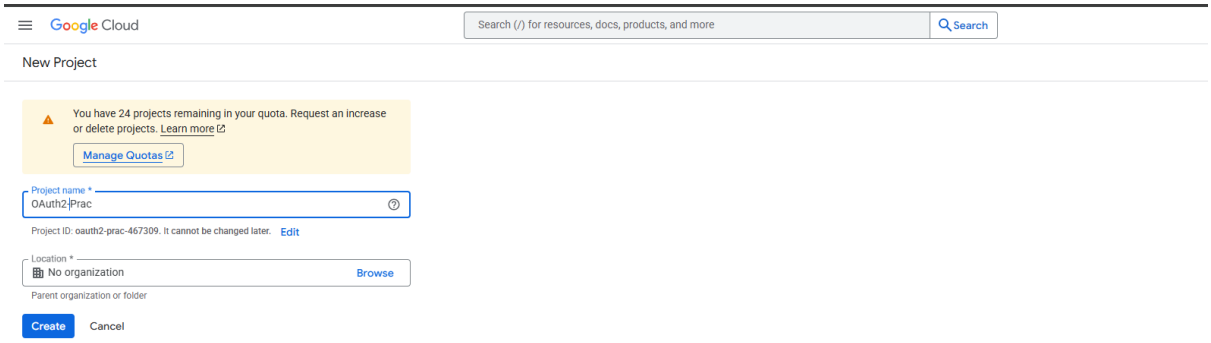
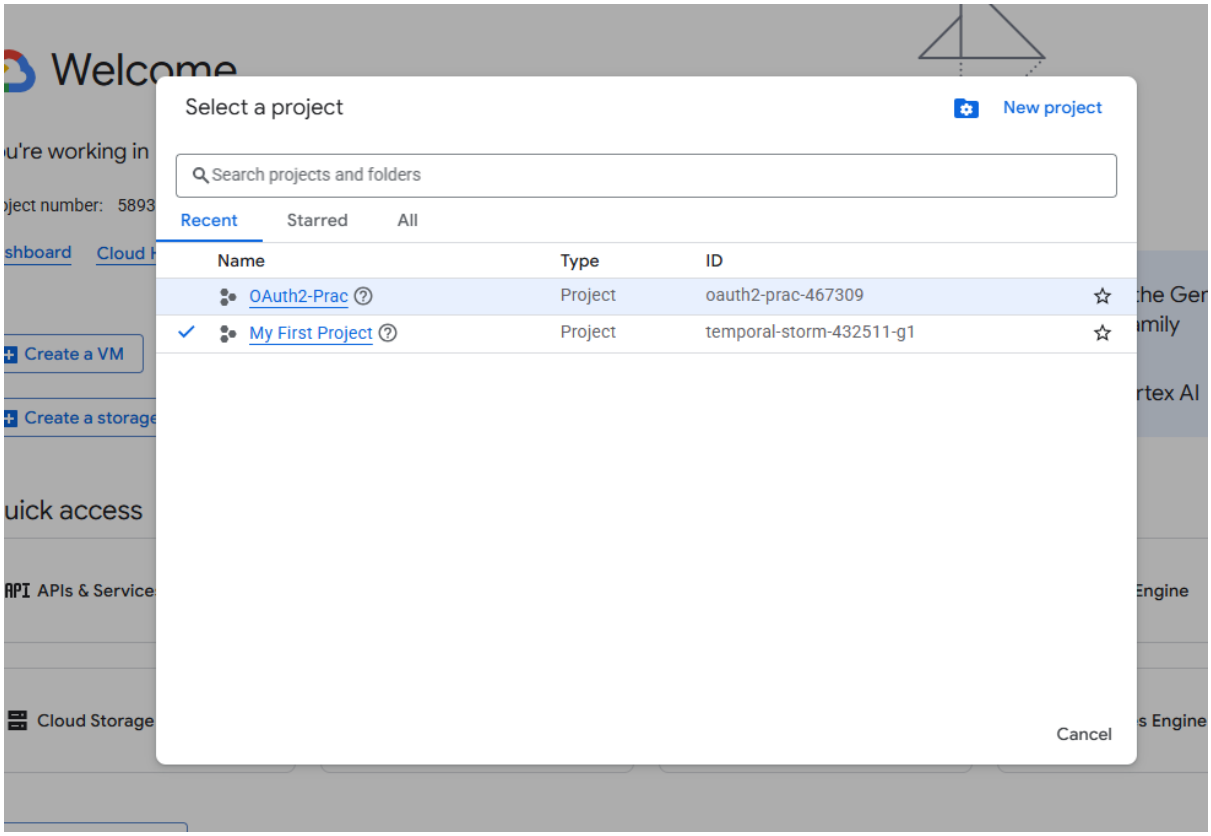
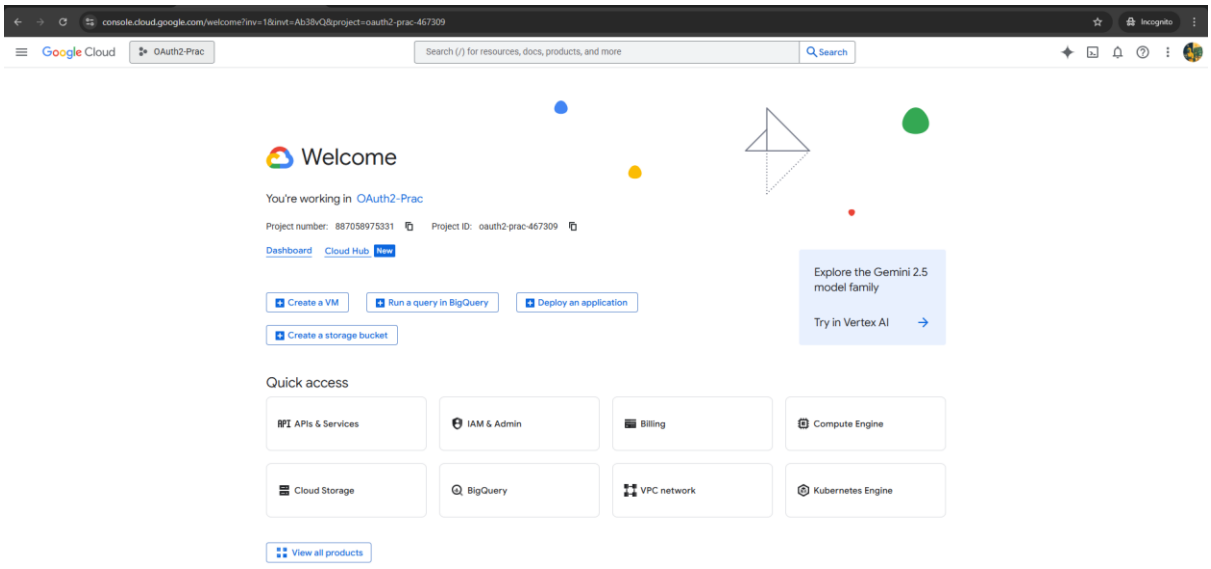
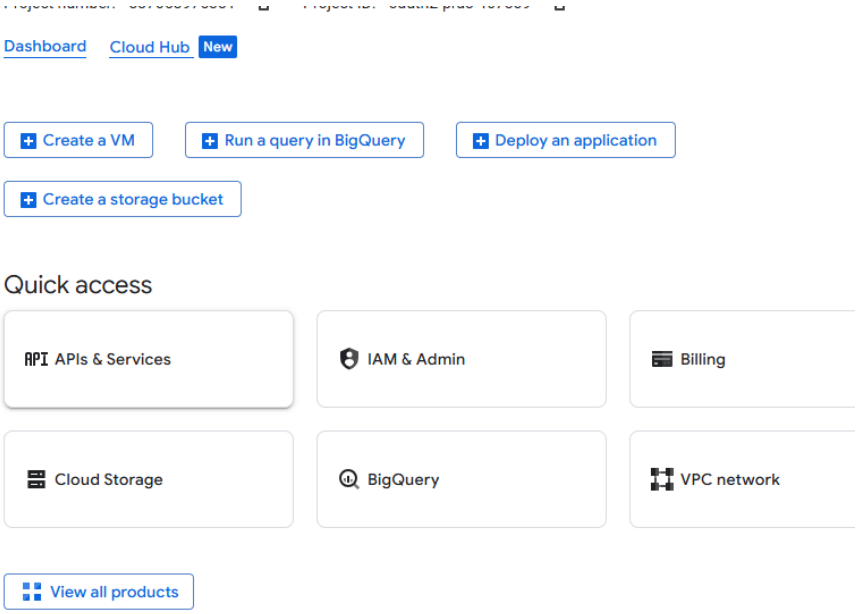
Practical 5 Suraj S Kaduvetti 40772240015 F015

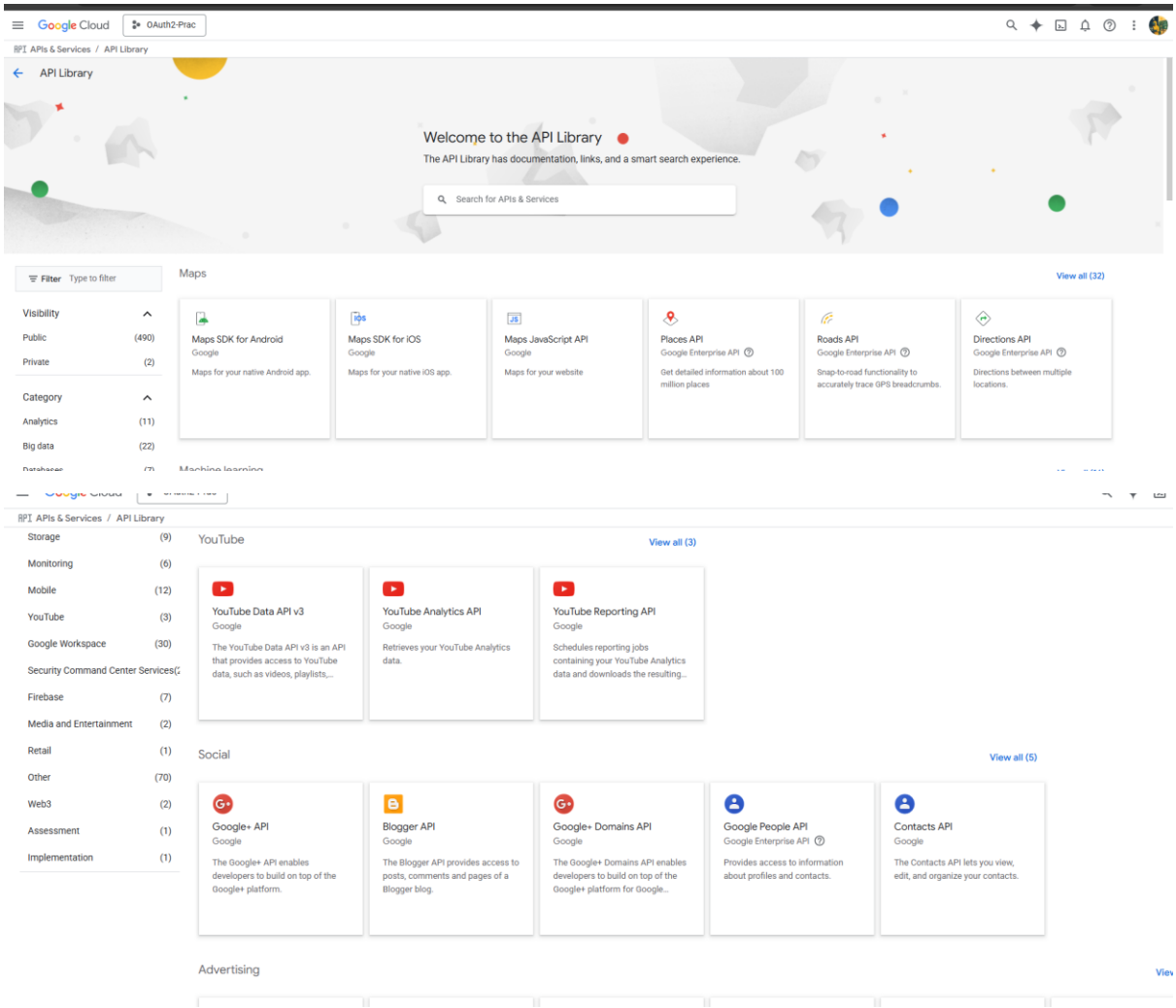


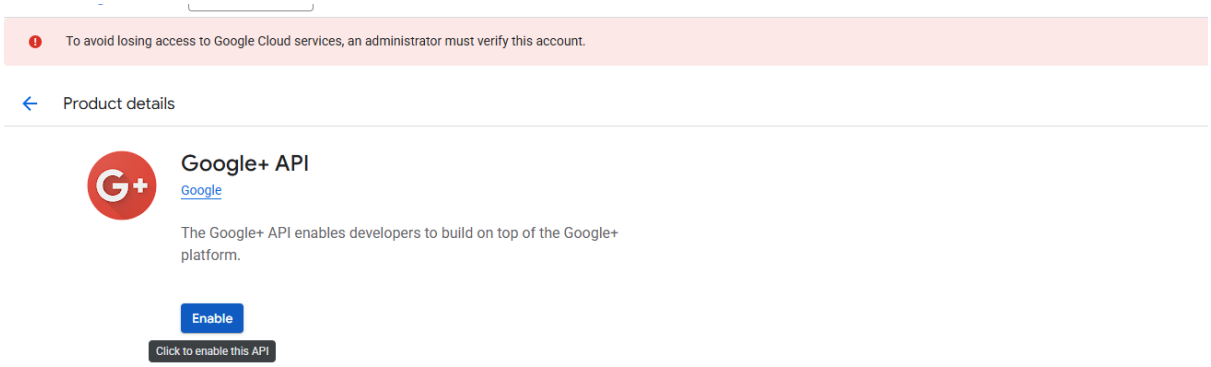

Welcome
Select a project
New project
Search projects and folders
Recent
Starred
All
Name
Type
ID
OAuth2-Prac
Project
oauth2-prac-467309
My First Project
Project
temporal-storm-432511-g1
Create a VM
Create a storage
Cloud Storage
Cancel

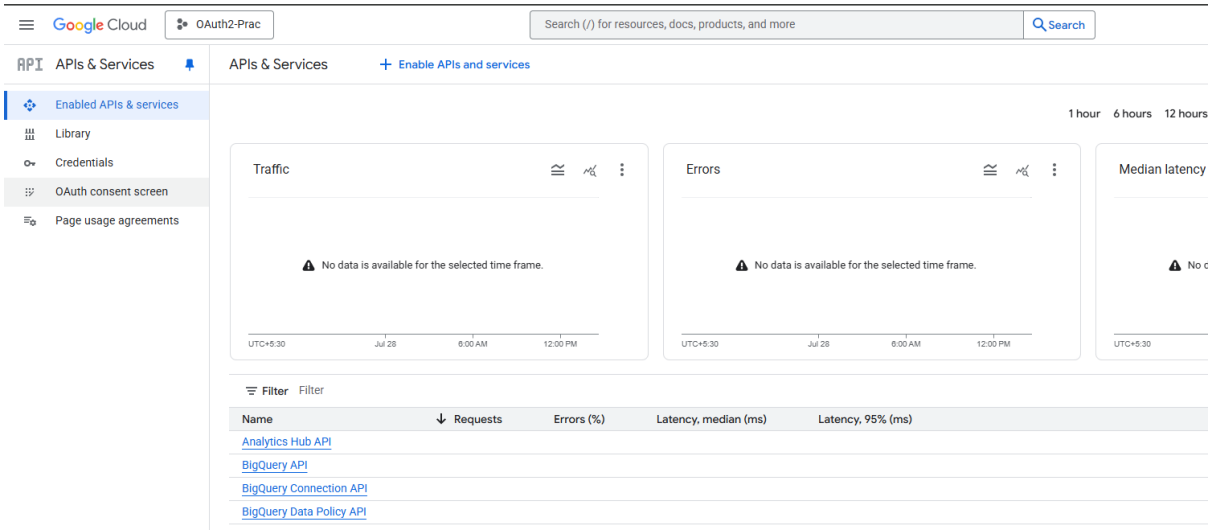

Google Cloud
OAuth2-Prac
Search (/) for resources, docs, products, and more
Search
Welcome
You're working in OAuth2-Prac
Project number: 887058975331
Project ID: oauth2-prac-467309
Dashboard
Cloud Hub
New
Create a VM
Run a query in BigQuery
Deploy an application
Create a storage bucket
Explore the Gemini 2.5 model family
Try in Vertex AI
Quick access
APIs & Services
IAM & Admin
Billing
Compute Engine
Cloud Storage
BigQuery
VPC network
Kubernetes Engine
View all products

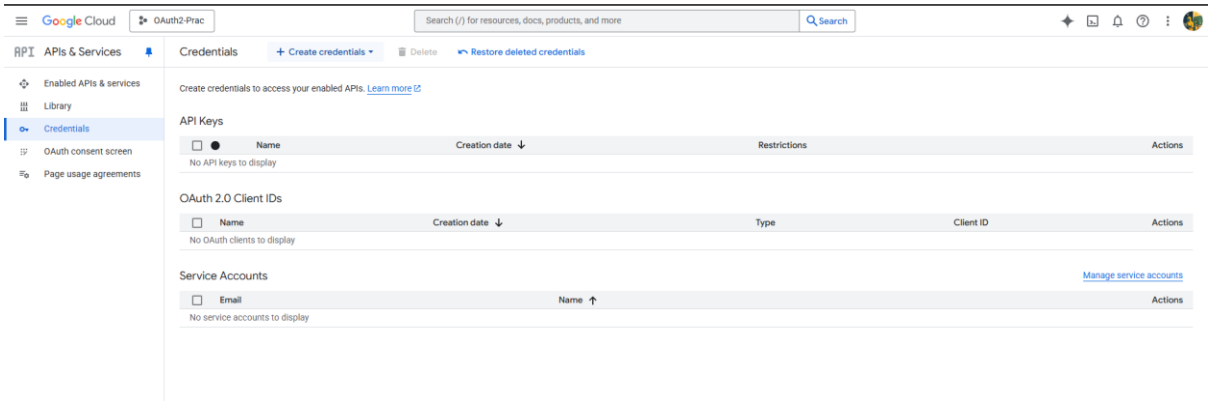

Dashboard
Cloud Hub
New
Create a VM
Run a query in BigQuery
Deploy an application
Create a storage bucket
Quick access
APIs & Services
IAM & Admin
Billing
Cloud Storage
BigQuery
VPC network
View all products


Google Cloud
OAuth2-Prac
APIs & Services / API Library
API Library
Welcome to the API Library
The API Library has documentation, links, and a smart search experience.
Search for APIs & Services
Filter Type to filter
Visibility
Public (490)
Private (2)
Category
Analytics (11)
Big data (22)
Maps
View all (32)
Maps SDK for Android
Google
Maps for your native Android app.
Maps SDK for iOS
Google
Maps for your native iOS app.
Maps JavaScript API
Google
Maps for your website
Places API
Google Enterprise API
Get detailed information about 100 million places
Roads API
Google Enterprise API
Snap-to-road functionality to accurately trace GPS breadcrumbs.
Directions API
Google Enterprise API
Directions between multiple locations.
Storage (9)
Monitoring (6)
Mobile (12)
YouTube (3)
Google Workspace (30)
Security Command Center Services(
Firebase (7)
Media and Entertainment (2)
Retail (1)
Other (70)
Web3 (2)
Assessment (1)
Implementation (1)
YouTube
View all (3)
YouTube Data API v3
Google
The YouTube Data API v3 is an API that provides access to YouTube data, such as videos, playlists,...
YouTube Analytics API
Google
Retrieves your YouTube Analytics data.
YouTube Reporting API
Google
Schedules reporting jobs containing your YouTube Analytics data and downloads the resulting...
Social
View all (5)
Google+ API
Google
The Google+ API enables developers to build on top of the Google+ platform.
Blogger API
Google
The Blogger API provides access to posts, comments and pages of a Blogger blog.
Google+ Domains API
Google
The Google+ Domains API enables developers to build on top of the Google+ platform for Google...
Google People API
Google Enterprise API
Provides access to information about profiles and contacts.
Contacts API
Google
The Contacts API lets you view, edit, and organize your contacts.
Advertising

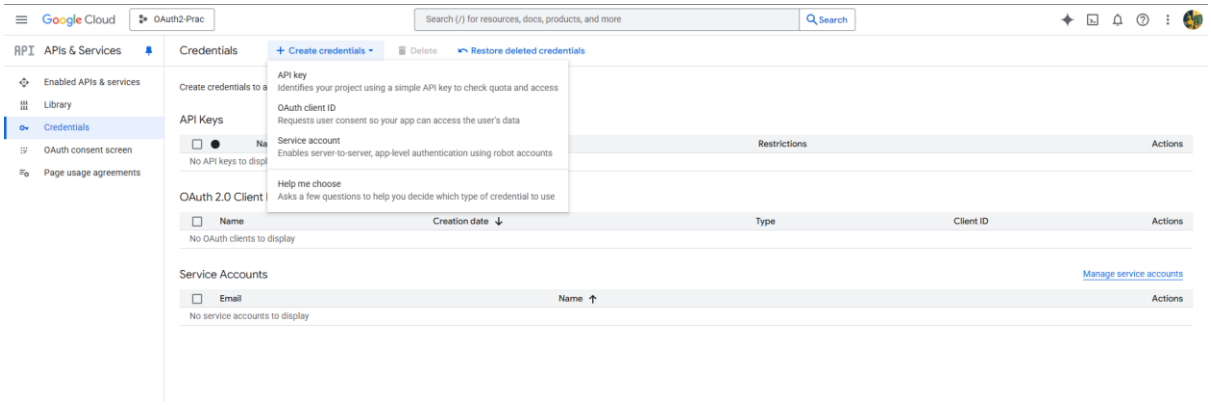
# Practical 5 Suraj S Kaduvetti 40772240015 F015

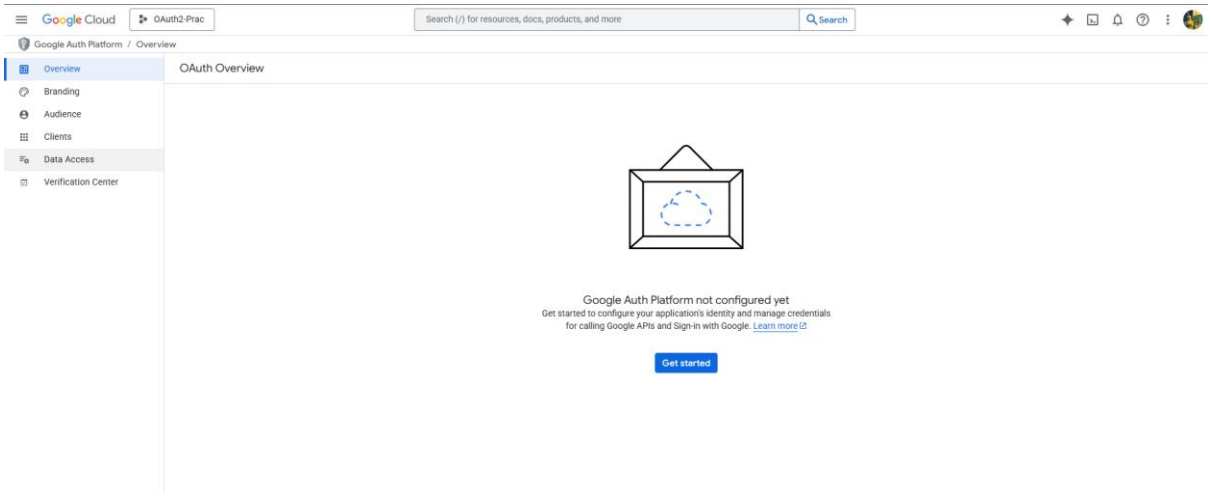


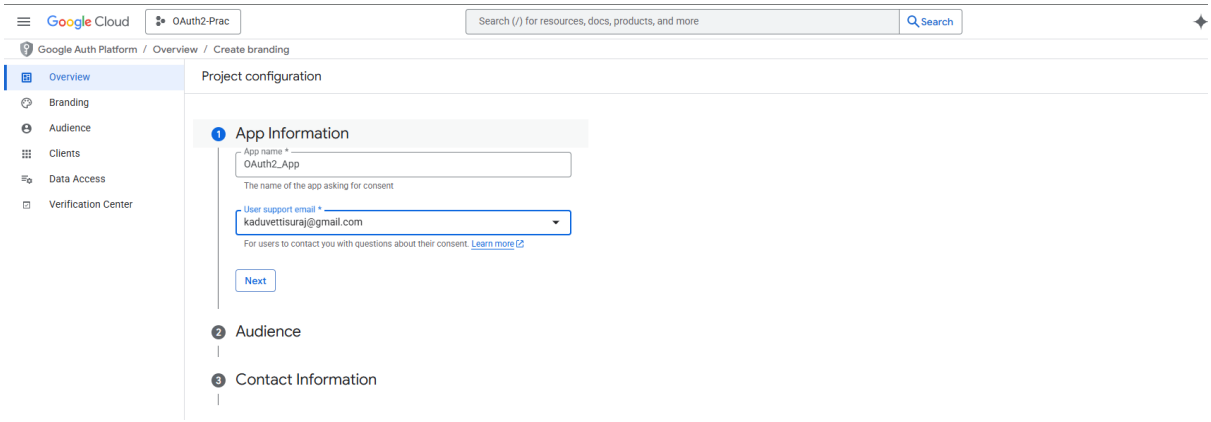



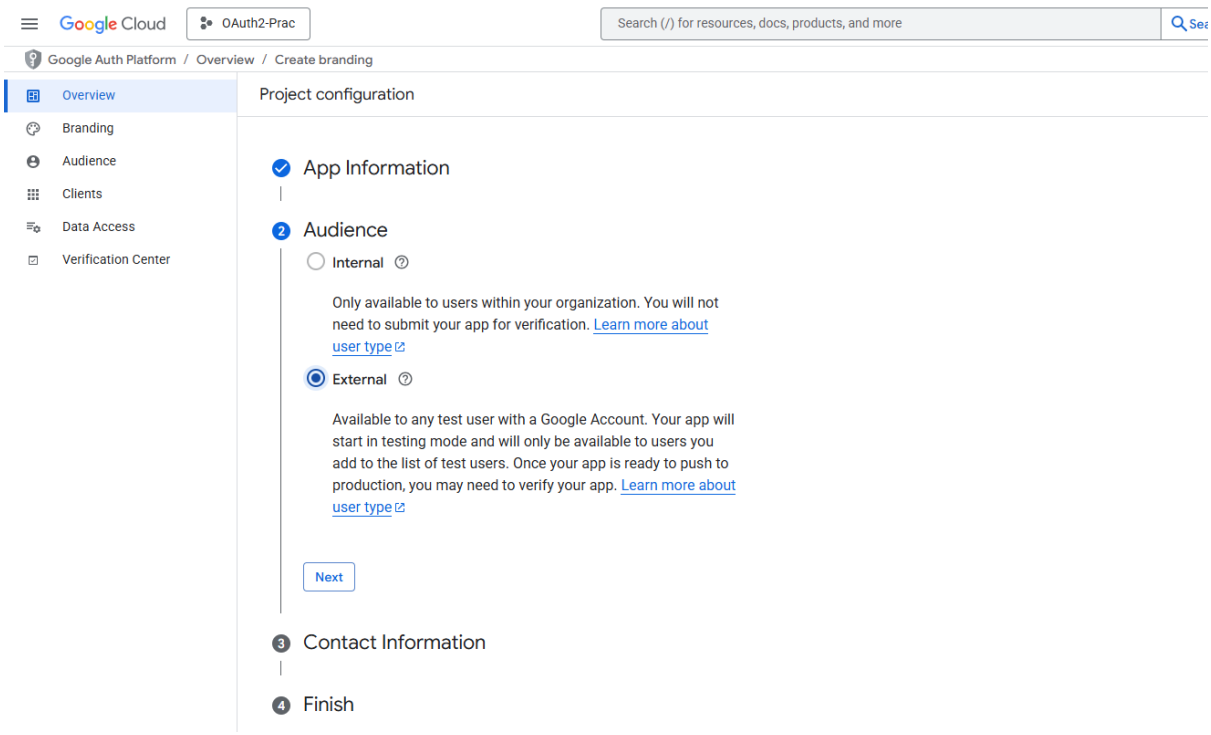
Practical 5 Suraj S Kaduvetti 40772240015 F015

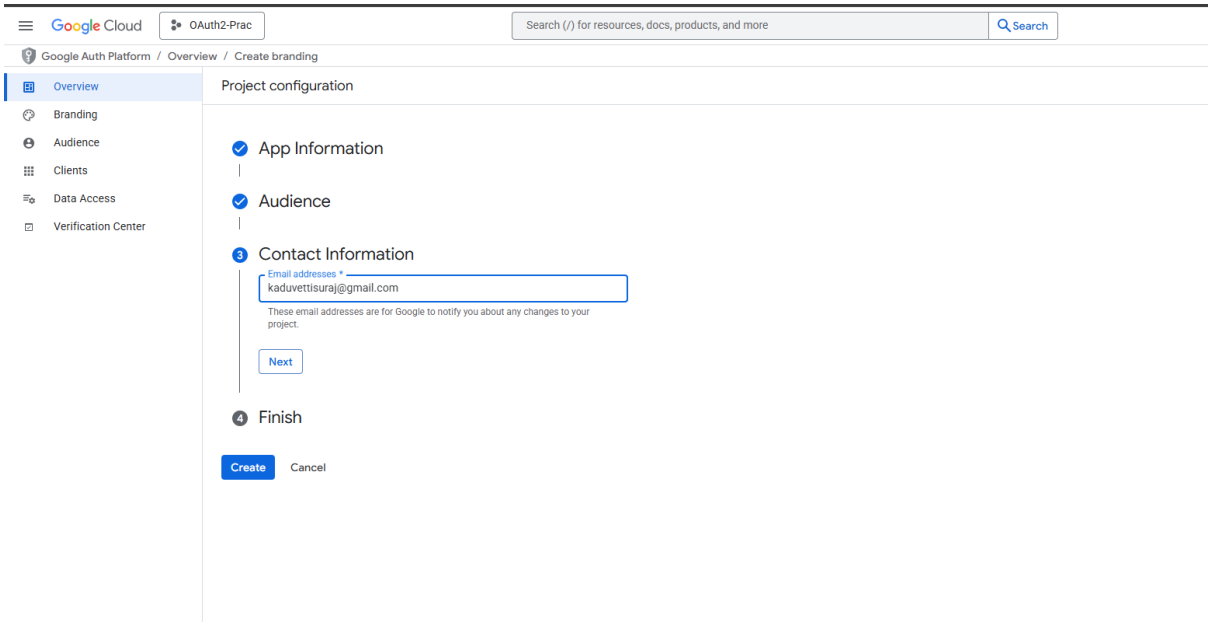


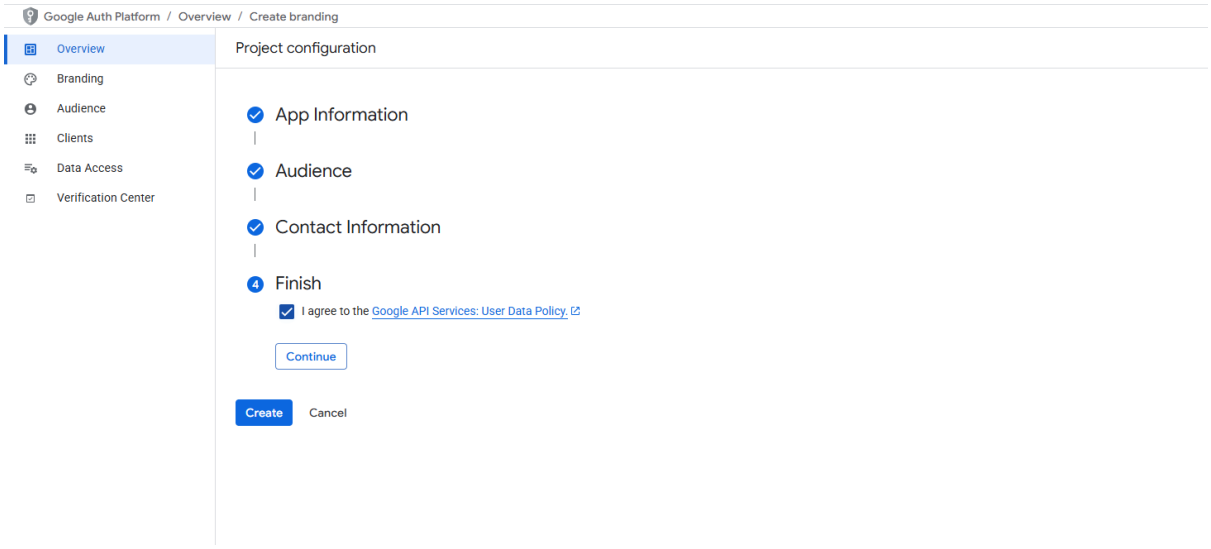



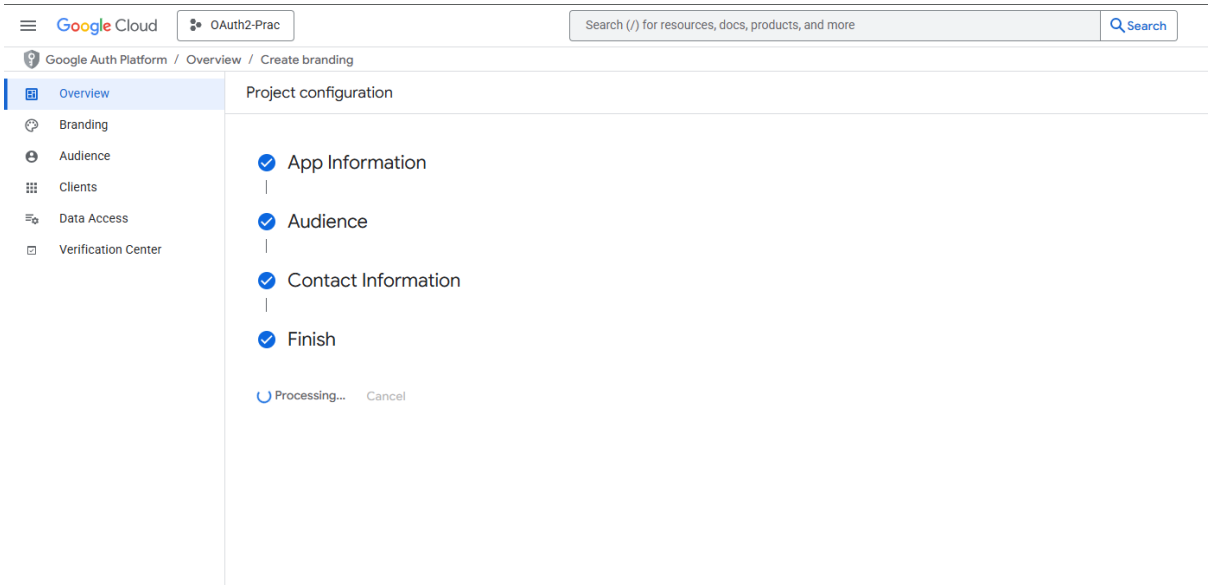
Practical 5 Suraj S Kaduvetti 40772240015 F015



# Practical 5 Suraj S Kaduvetti 40772240015 F015



Practical 5 Suraj S Kaduvetti 40772240015 F015

Google Cloud | OAuth2-Prac | Search (/) for resources, docs, products,

Google Auth Platform / Clients / Client: 887058975331-2eho560gb0ghqmaqbpdh08ctmigoq89h.apps.googleusercontent.com

- Overview
- Branding
- Audience
- Clients
- Data Access
- Verification Center

Client ID for Web application | Delete

Name *
Spring Boot OAuth App

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your OAuth consent screen as authorized domains.

## Authorized JavaScript origins

For use with requests from a browser

URIs 1 *
https://localhost:8080

+ Add URI

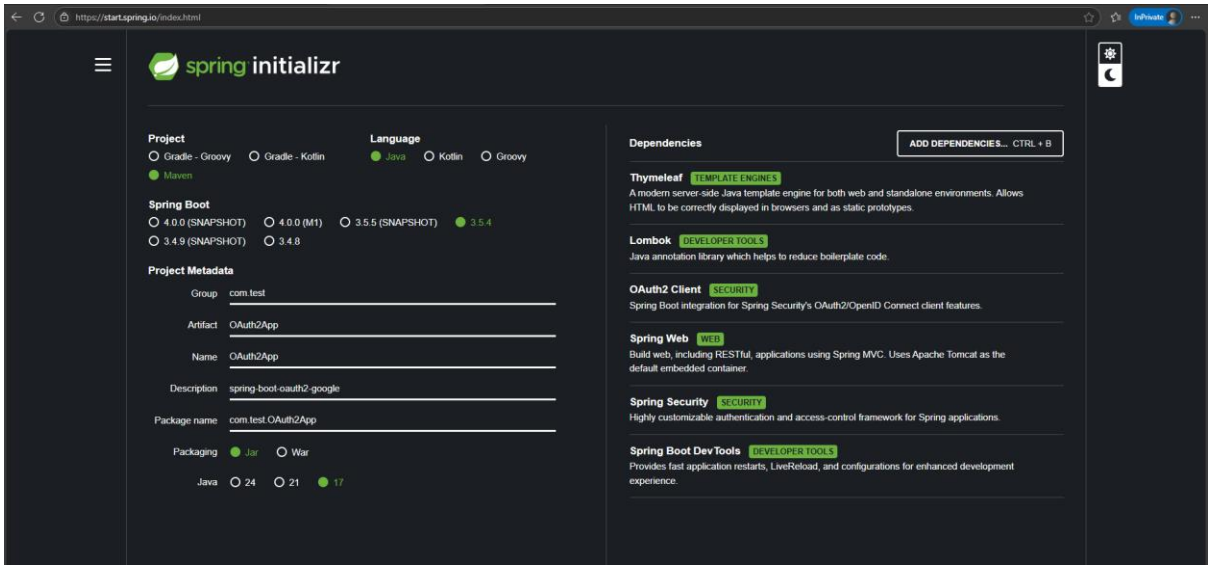
## Authorized redirect URIs

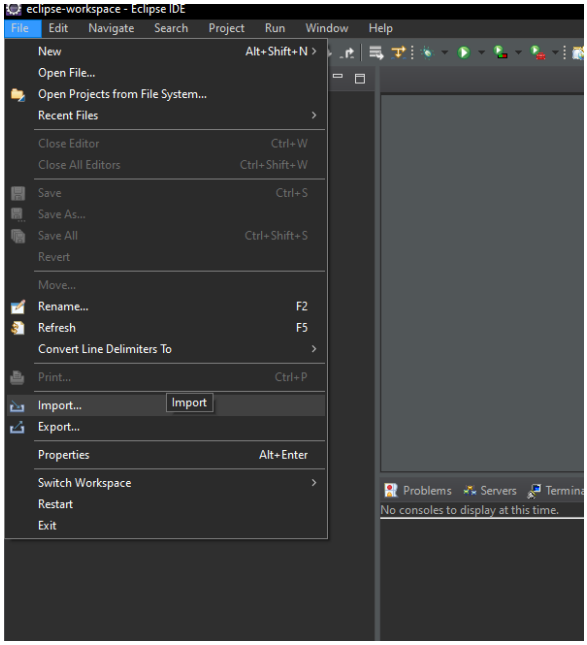
For use with requests from a web server

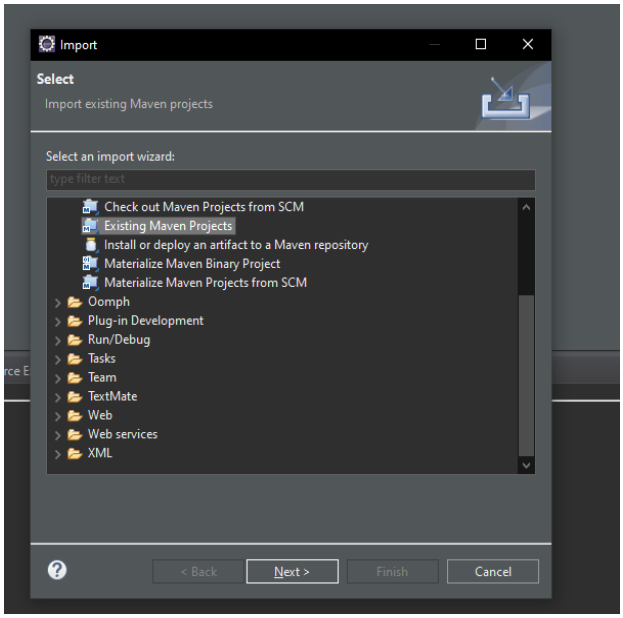
URIs 1 *
http://localhost:8080/login/oauth2/code/google

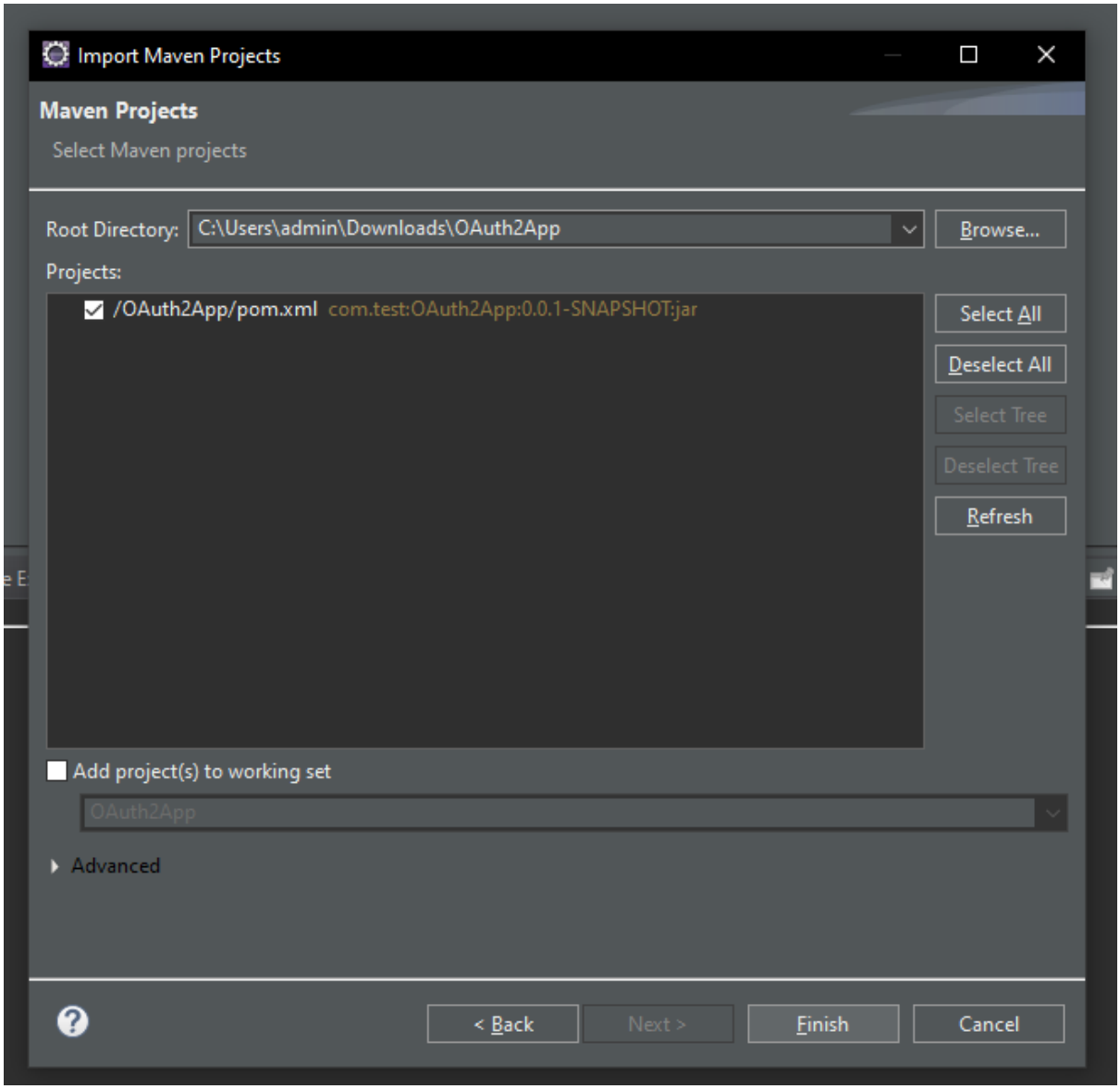
+ Add URI

Note: It may take 5 minutes to a few hours for settings to take effect

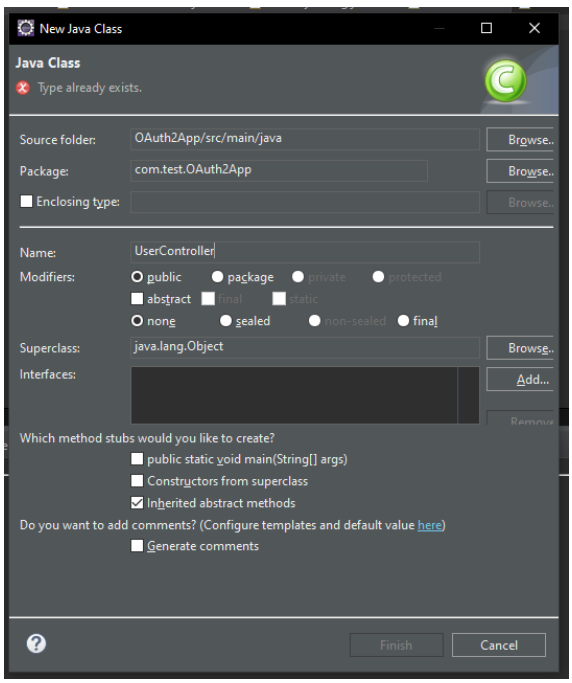


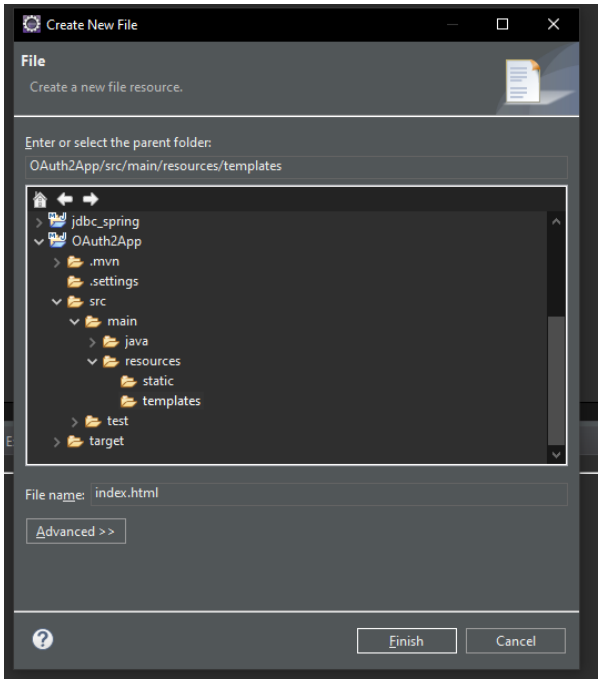

eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help
New Alt+Shift+N
Open File...
Open Projects from File System...
Recent Files
Close Editor Ctrl+W
Close All Editors Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Import... Import
Export...
Properties Alt+Enter
Switch Workspace
Restart
Exit
Problems Servers
No consoles to display at this time.

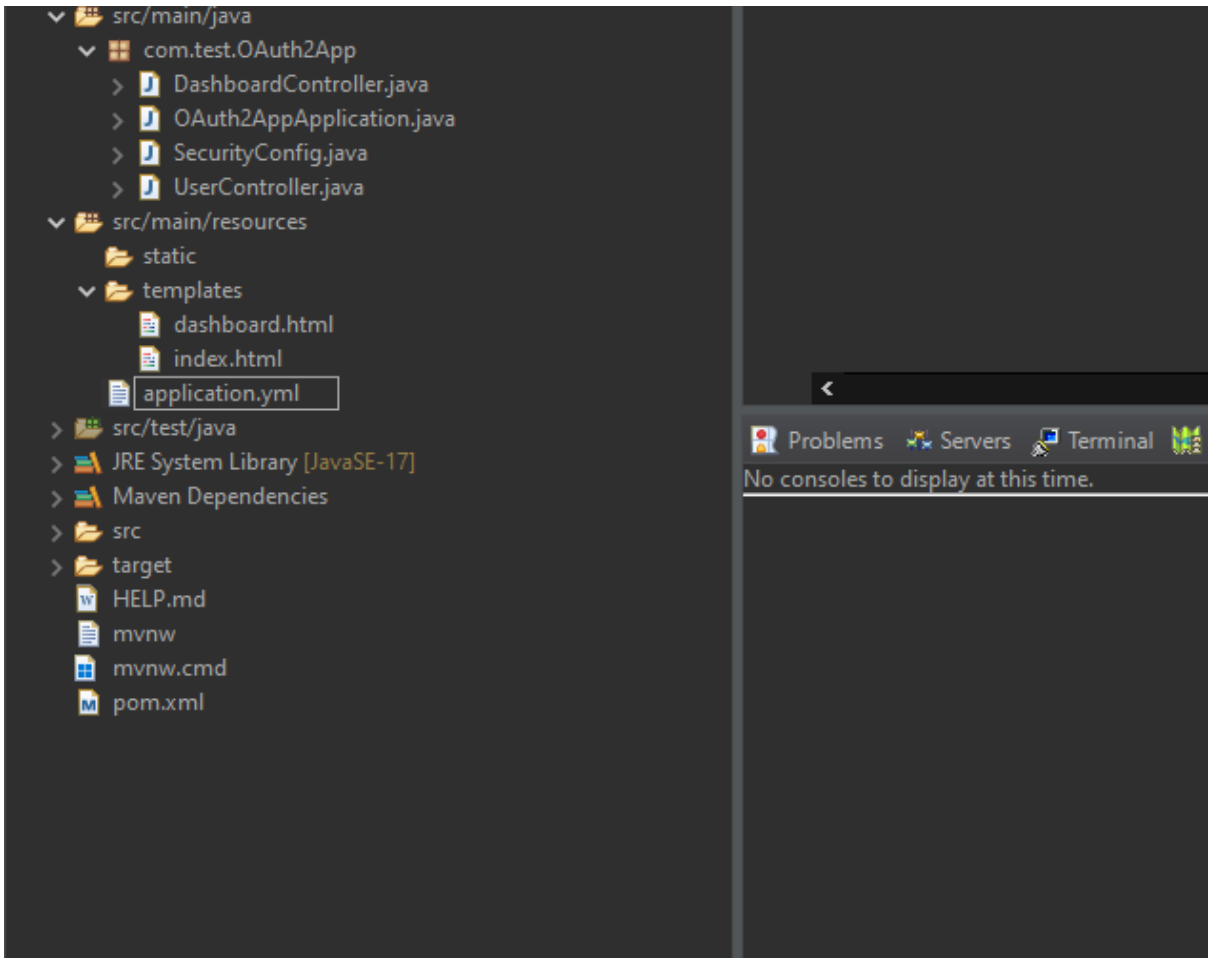

Import
Select
Import existing Maven projects
Select an import wizard:
type filter text
Check out Maven Projects from SCM
Existing Maven Projects
Install or deploy an artifact to a Maven repository
Materialize Maven Binary Project
Materialize Maven Projects from SCM
Oomph
Plug-in Development
Run/Debug
Tasks
Team
TextMate
Web
Web services
XML
< Back
Next >
Finish
Cancel


Import Maven Projects
Maven Projects
Select Maven projects
Root Directory: C:\Users\admin\Downloads\OAuth2App
Browse...
Projects:
/OAuth2App/pom.xml com.test:OAuth2App:0.0.1-SNAPSHOT:jar
Select All
Deselect All
Select Tree
Deselect Tree
Refresh
Add project(s) to working set
OAuth2App
Advanced
< Back
Next >
Finish
Cancel

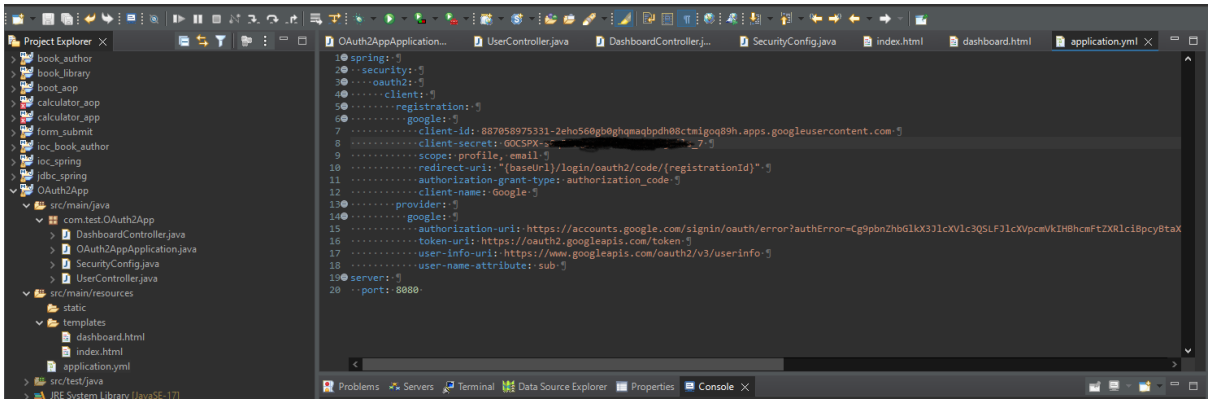
Practical 5 Suraj S Kaduvetti 40772240015 F015

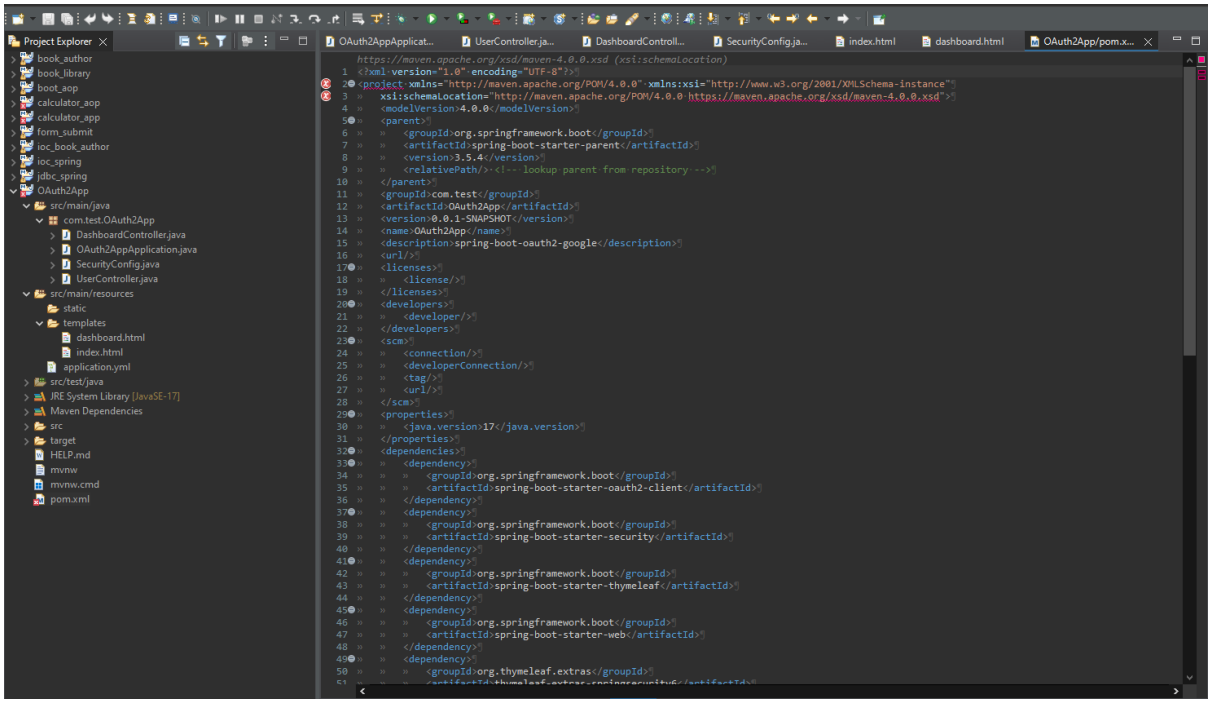


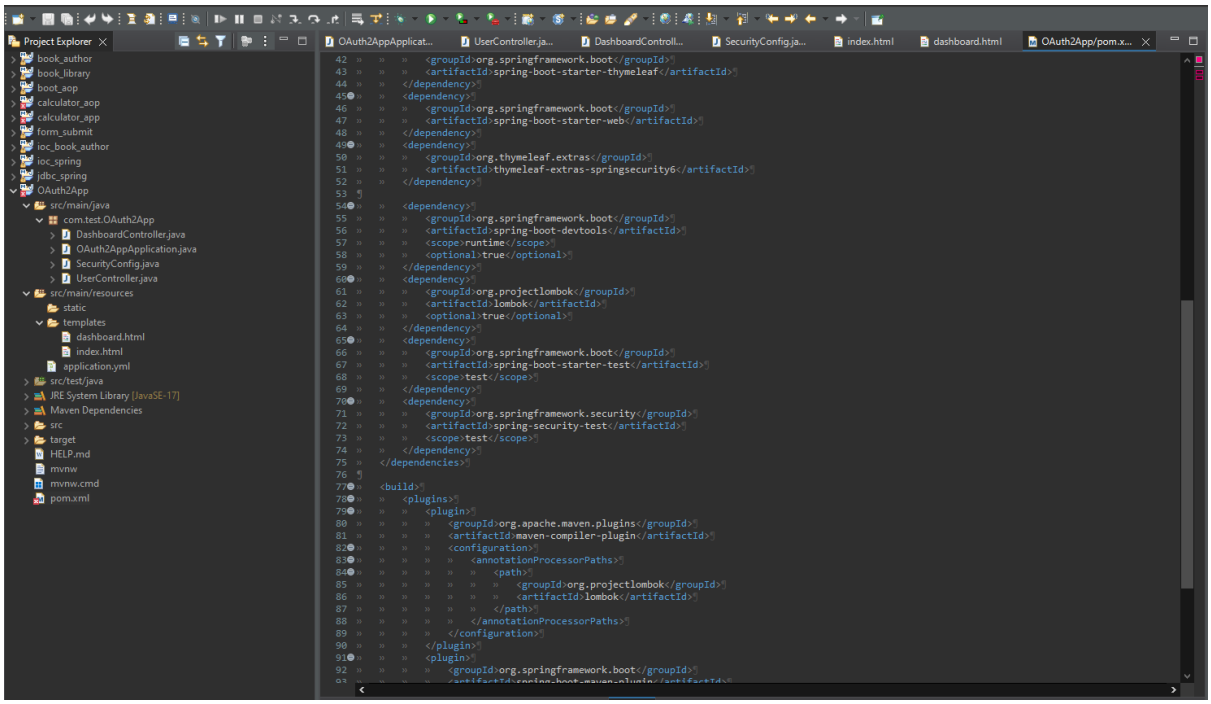



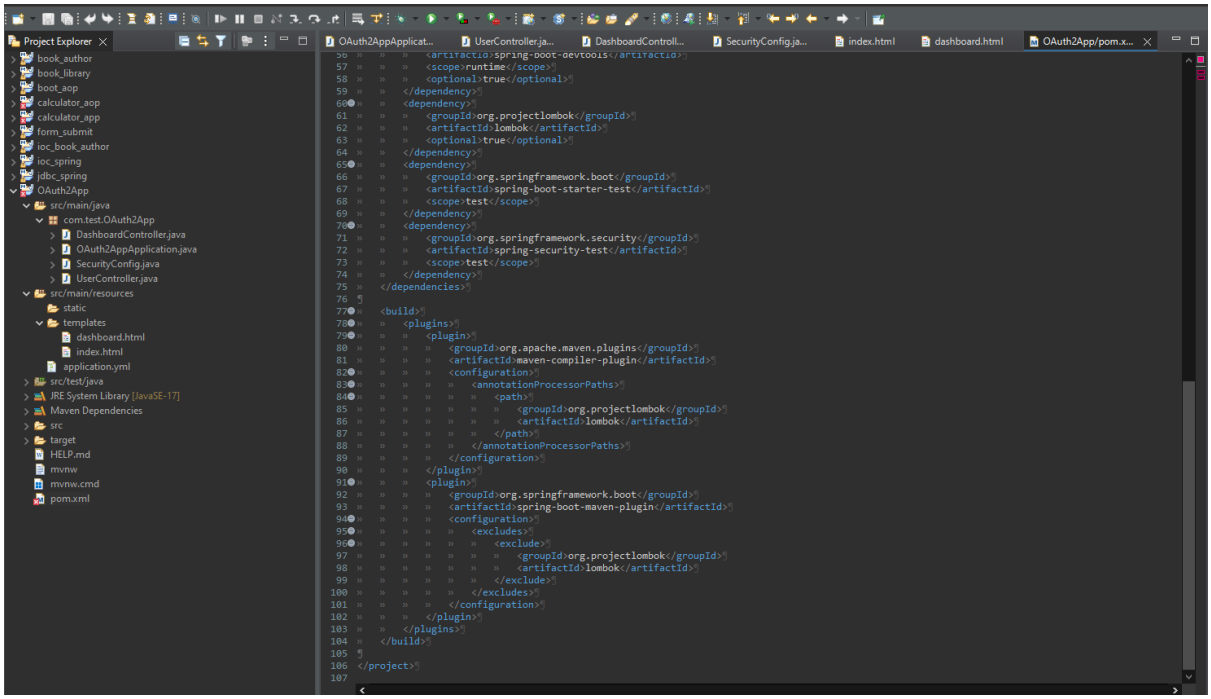
Practical 5 Suraj S Kaduvetti 40772240015 F015

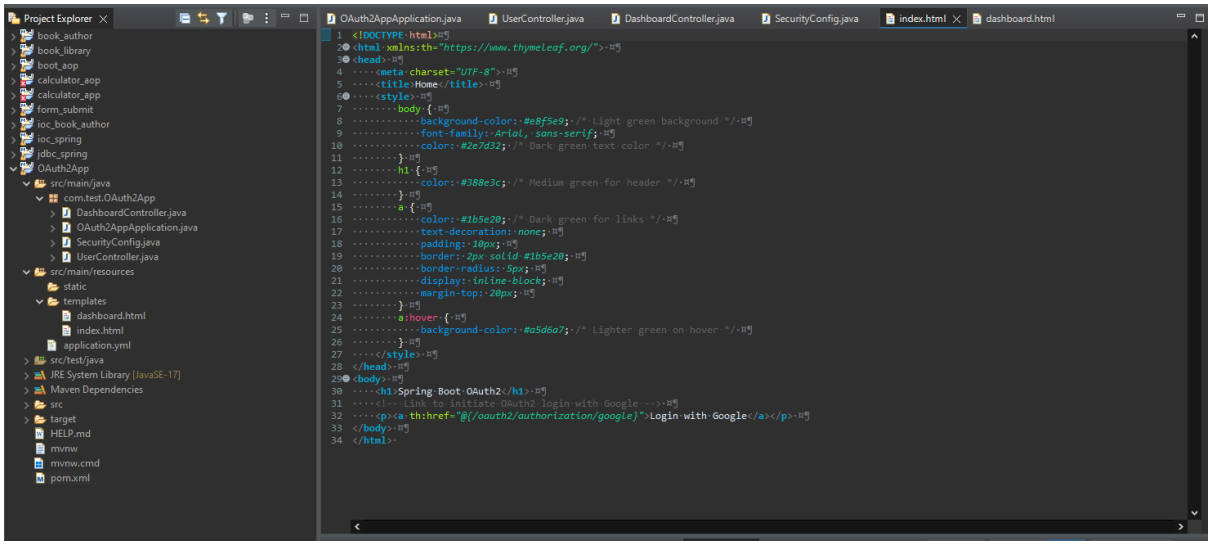


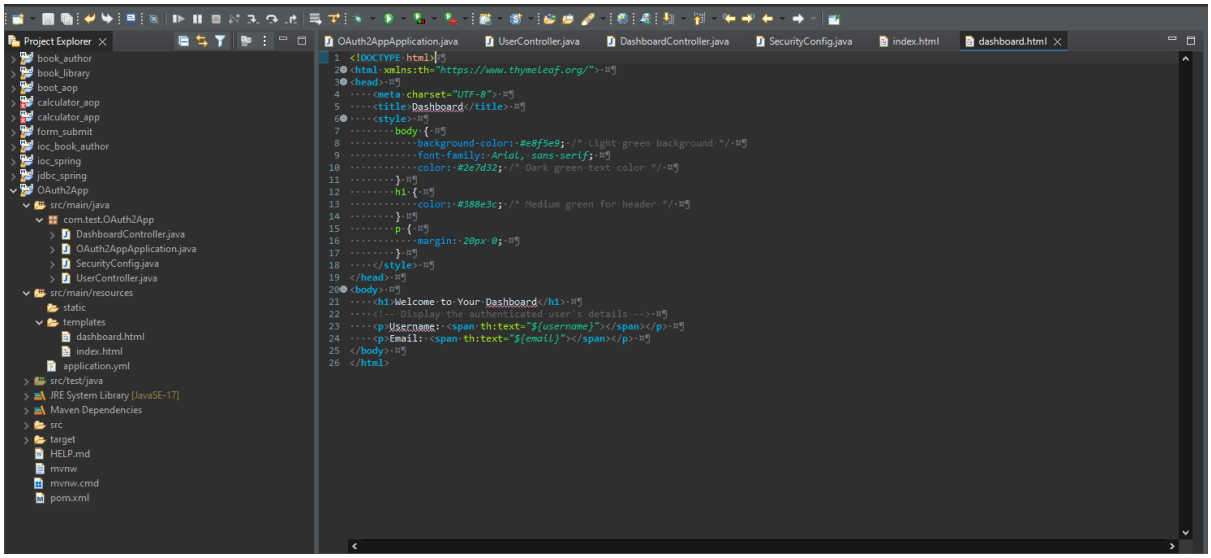



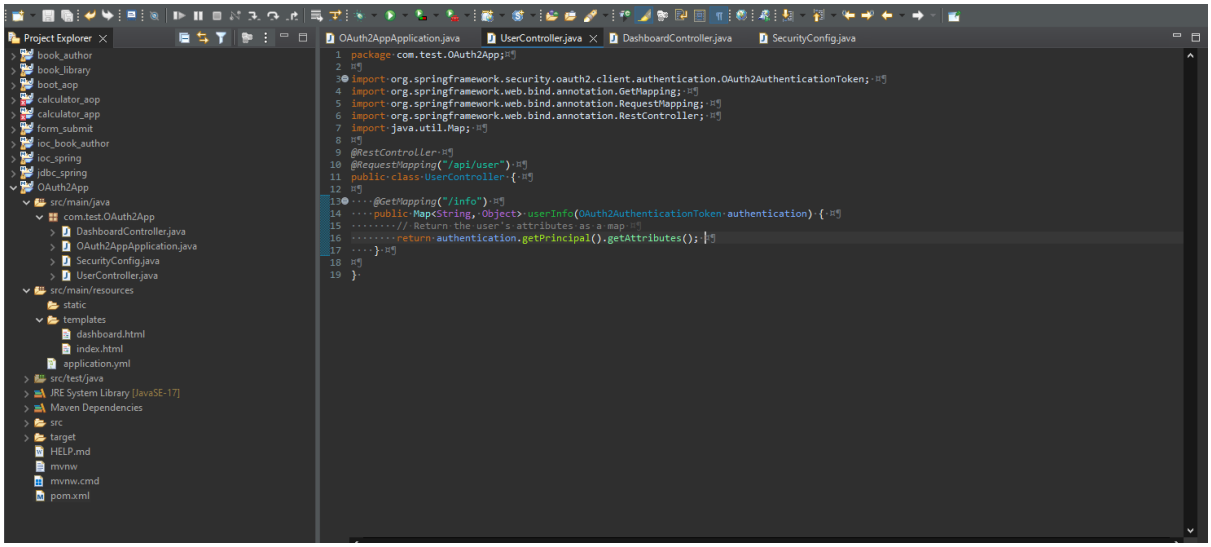
Practical 5 Suraj S Kaduvetti 40772240015 F015



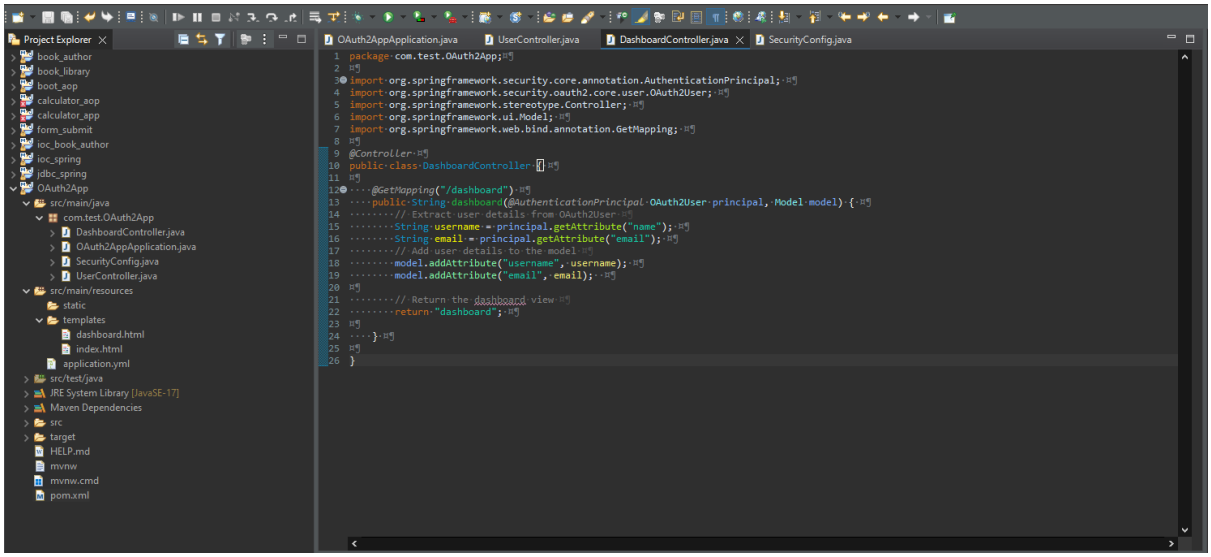



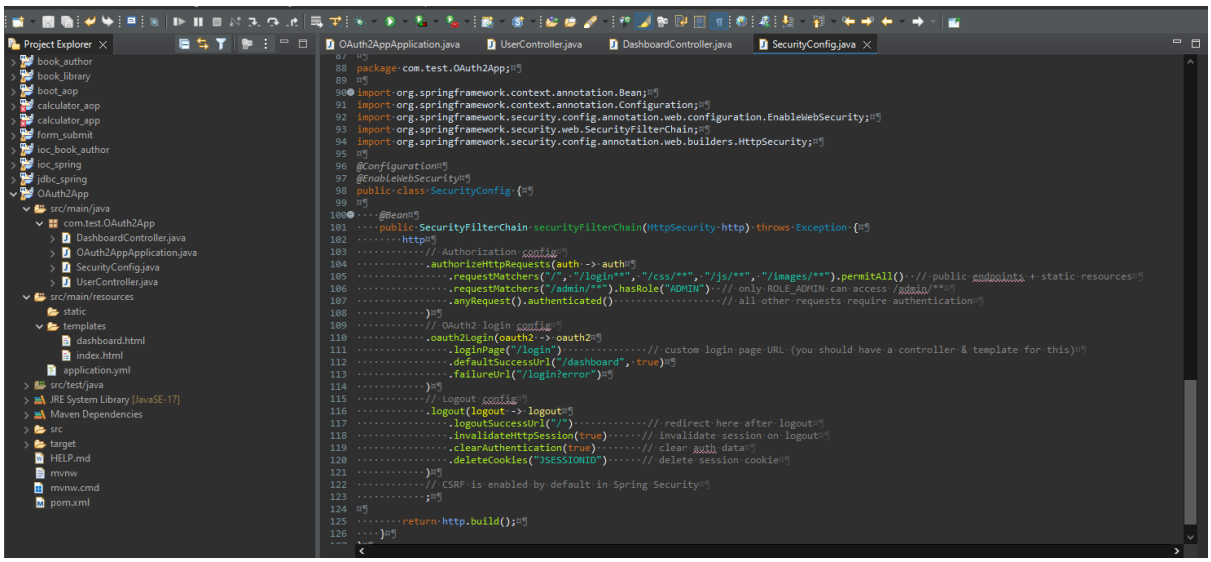
Practical 5 Suraj S Kaduvetti 40772240015 F015



```
package com.test.OAuth2App;

import org.springframework.security.oauth2.client.authentication.OAuth2AuthenticationToken;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import java.util.Map;

@RestController
@RequestMapping("/api/user")
public class UserController {

    @GetMapping("/info")
    public Map<String, Object> userInfo(OAuth2AuthenticationToken authentication) {
        // Return the user's attributes as a map
        return authentication.getPrincipal().getAttributes();
    }

}
```

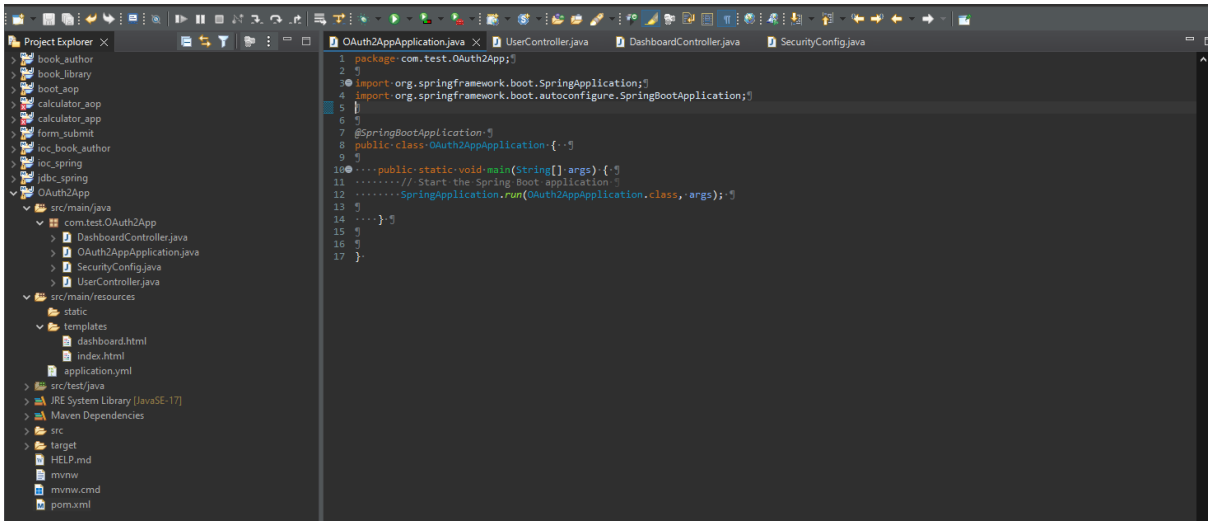


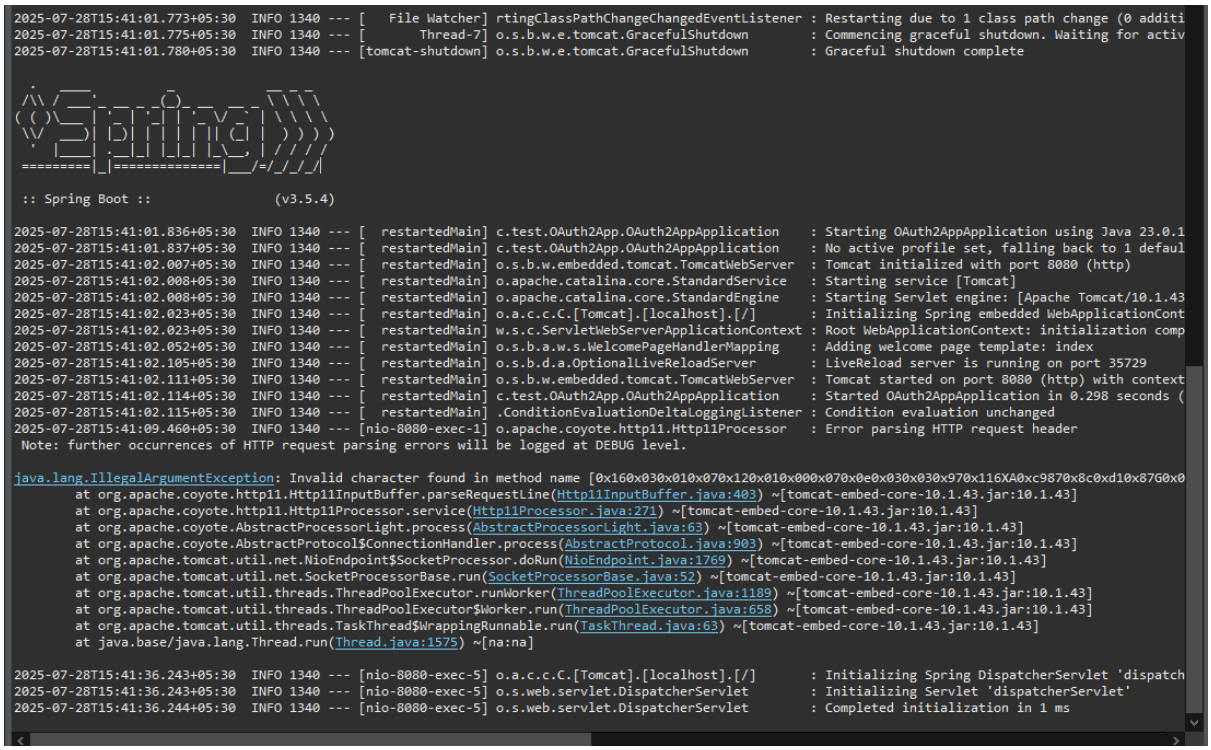
```
package com.test.OAuth2App;

import org.springframework.security.core.annotation.AuthenticationPrincipal;
import org.springframework.security.oauth2.core.user.OAuth2User;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class DashboardController {

    @GetMapping("/dashboard")
    public String dashboard(@AuthenticationPrincipal OAuth2User principal, Model model) {
        // Extract user details from OAuth2User
        String username = principal.getAttribute("name");
        String email = principal.getAttribute("email");
        // Add user details to the model
        model.addAttribute("username", username);
        model.addAttribute("email", email);

        // Return the dashboard view
        return "dashboard";

    }

}
```



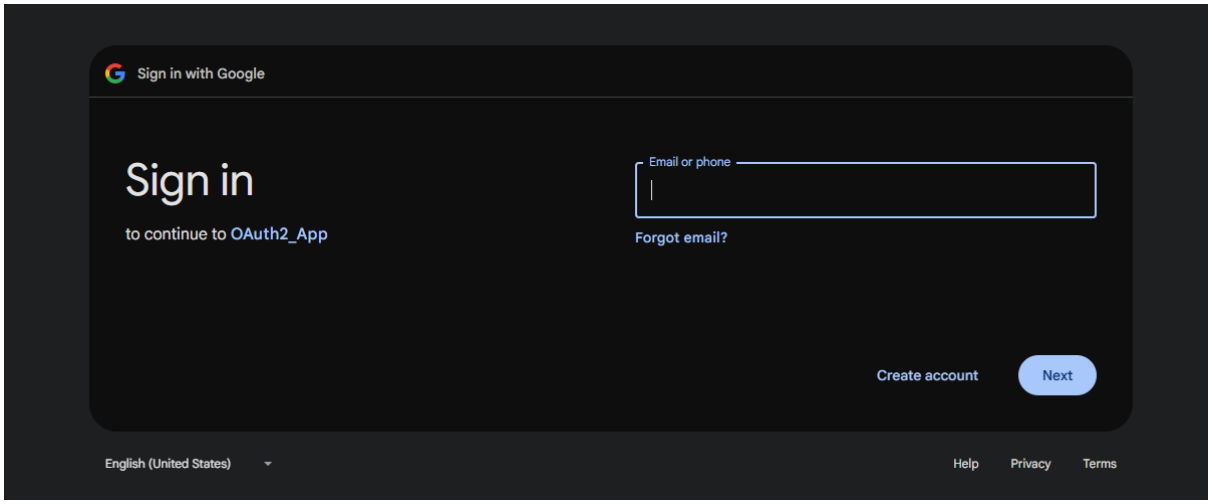
```
package com.test.OAuth2App;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.web.SecurityFilterChain;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;

@Configuration
@EnableWebSecurity
public class SecurityConfig {

    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
        http
            // Authorization config
            .authorizeHttpRequests(auth -> auth
                .requestMatchers("/","/login**","/css/**","/js/**","/images/**").permitAll() // public endpoints + static resources
                .requestMatchers("/admin/**").hasRole("ADMIN") // only ROLE_ADMIN can access /admin/**
                .anyRequest().authenticated()                  // all other requests require authentication
            )
            // OAuth2 login config
            .oauth2Login(oauth2 -> oauth2
                .loginPage("/login")                // custom login page URL (you should have a controller & template for this)
                .defaultSuccessUrl("/dashboard", true)
                .failureUrl("/login?error")
            )
            // Logout config
            .logout(logout -> logout
                .logoutSuccessUrl("/")              // redirect here after logout
                .invalidateHttpSession(true)        // invalidate session on logout
                .clearAuthentication(true)          // clear auth data
                .deleteCookies("JSESSIONID")        // delete session cookie
            )
            // CSRF is enabled by default in Spring Security
            ;

        return http.build();
    }
```

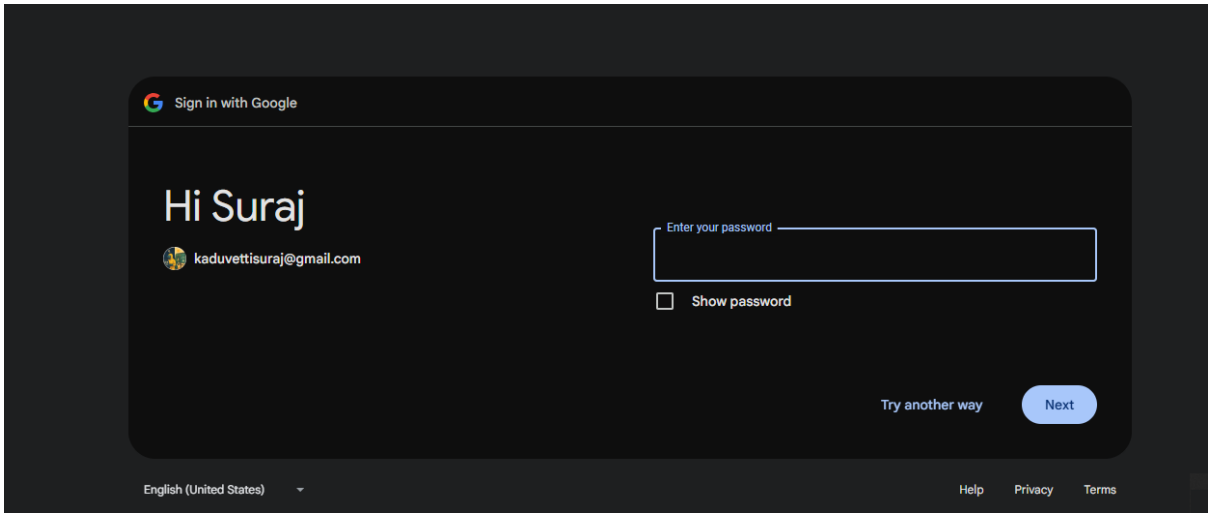
Practical 5 Suraj S Kaduvetti 40772240015 F015

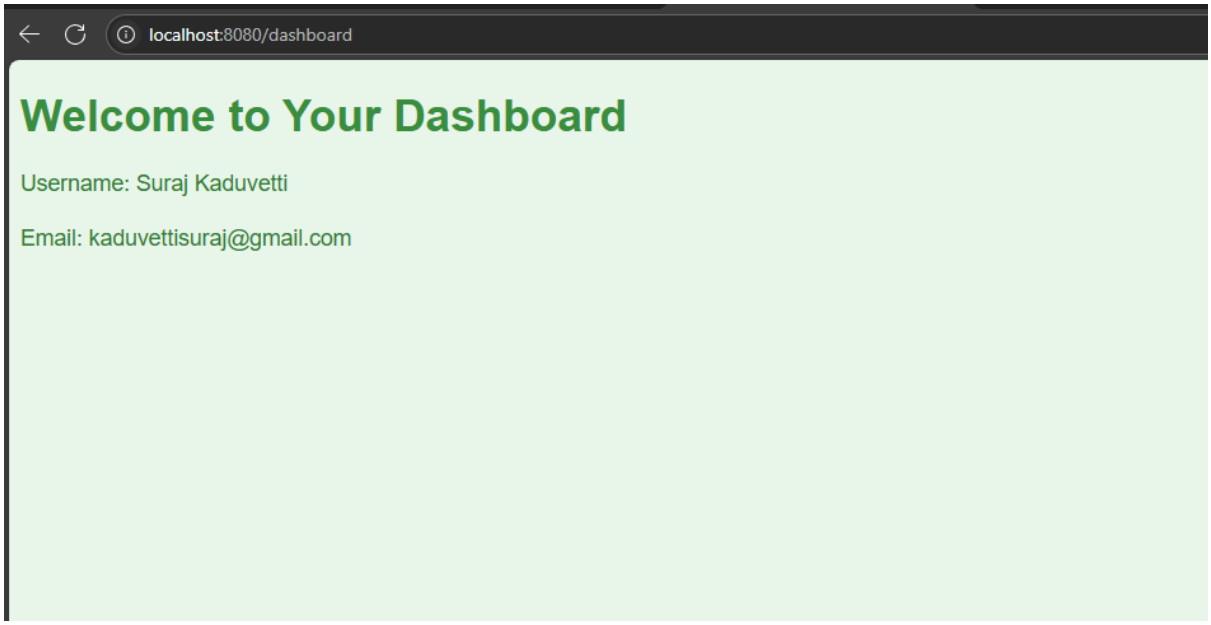


localhost:8080
Spring Boot OAuth2
Login with Google


Sign in with Google
Sign in
to continue to OAuth2_App
Email or phone
Forgot email?
Create account
Next
English (United States)
Help
Privacy
Terms


Sign in with Google
Hi Suraj
kaduvettisuraj@gmail.com
Enter your password
Show password
Try another way
Next
English (United States)
Help
Privacy
Terms


localhost:8080/dashboard
Welcome to Your Dashboard
Username: Suraj Kaduvetti
Email: kaduvettisuraj@gmail.com