

**SEB-312 Mobile Application Development**

**LAB # 01**

**LAB Title**

|  |
| --- |
| Install Visual Studio Code and also install flutter and dart extensions. Create a first application of Hello World. Exploring project structure and main files. Create a basic counter app on Flutter. Writing simple Dart programs. |

**Assessment of CLO: 03, PLO: 05**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name:** |  | | |
| **Roll No.** |  | | |
| **Semester** |  | **Session** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S. No.** | **Perf. Level**  **Criteria** | **Excellent**  **(2.5)** | **Good**  **(2)** | **Satisfactory**  **(1.5)** | **Needs Improvement**  **(0 ~ 1)** | **Marks Obtained** |
| **1** | Project Execution & Implementation | Fully functional, optimized, and well-structured. | Minor errors, mostly functional. | Some errors, requires guidance. | Major errors, non-functional, or not Performed. |  |
| **2** | Results & Debugging  Or Troubleshooting | Accurate results with effective debugging  Or Troubleshooting. | Mostly correct, some debugging Or Troubleshooting needed. | Partial results, minimal debugging  Or Troubleshooting. | Incorrect results, no debugging Or Troubleshooting, or not attempted. |  |
| **3** | Problem-Solving & Adaptability  (VIVA) | Creative approach, efficiently solves challenges. | Adapts well, minor struggles. | Some adaptability, needs guidance. | Lacks innovation or no innovation, unable to solve problems. |  |
| **4** | Report Quality & Documentation | Clear, structured, with detailed visuals. | Mostly clear, minor gaps. | Some clarity issues, missing details. | Poorly structured, lacks clarity, or not submitted. |  |
| **Total Marks Obtained Out of 10** | | | | | |  |

**Experiment evaluated by**

|  |  |  |  |
| --- | --- | --- | --- |
| **Instructor’s Name** | **Sidra Khatoon** | | |
| **Date** |  | **Signature** |  |

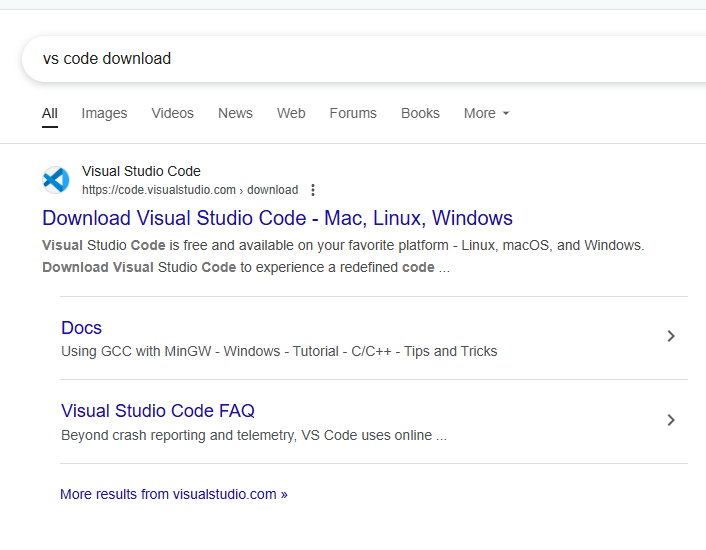
# **Objective**

The purpose of this lab session is installing visual studio code. Also install Flutter extension is VS code. Create a first application of Hello World.

**Labs Descriptions**

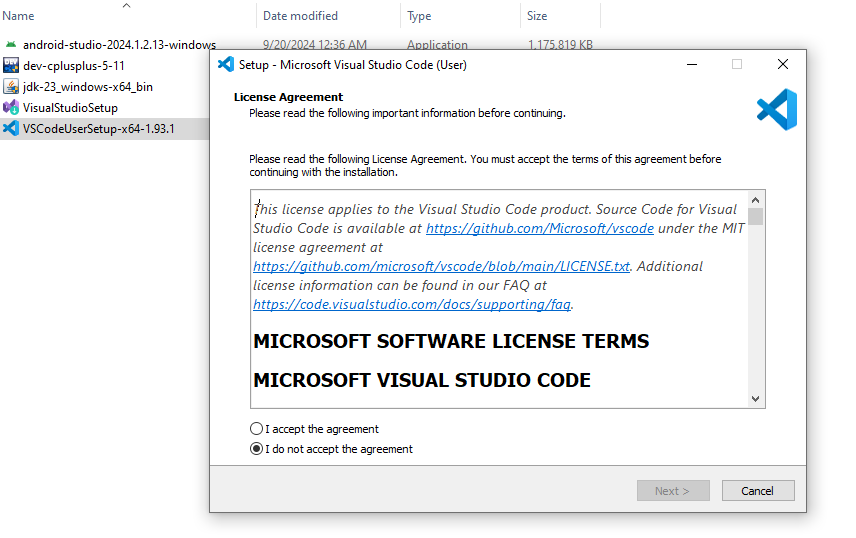
**Download VS code:**

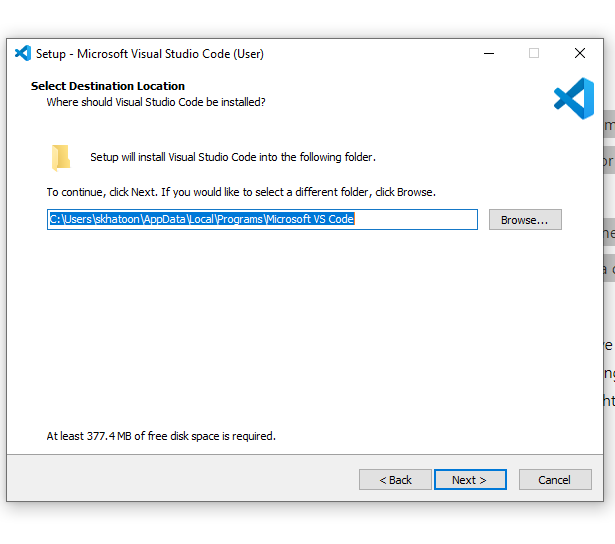
You can download VS code form their official website.

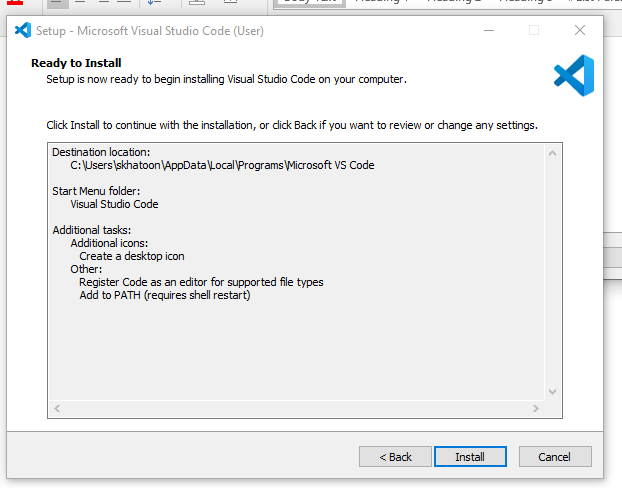


**Installation of VS code:**

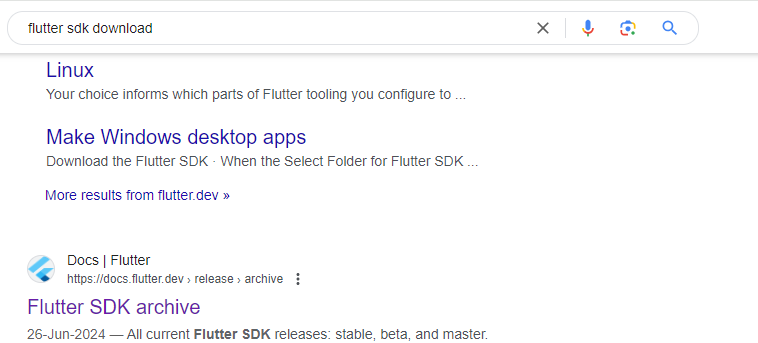
After download you can install the VS code. To install visual studio code run its setup file. Next move to the further steps for installation.

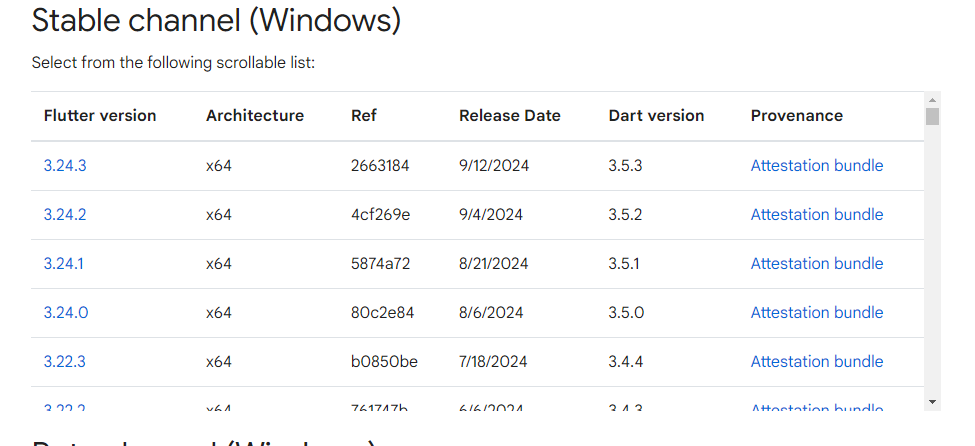




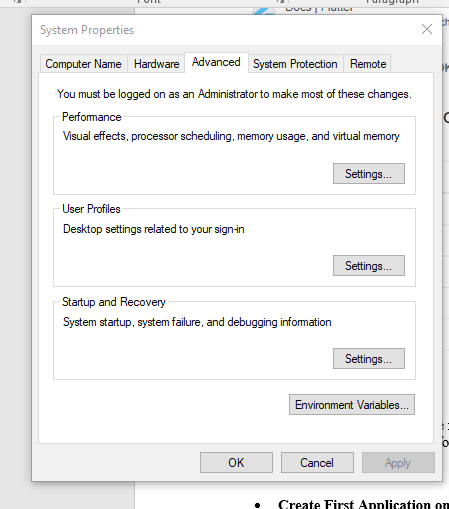


After VS code Installation process, we have download flutter extension files.

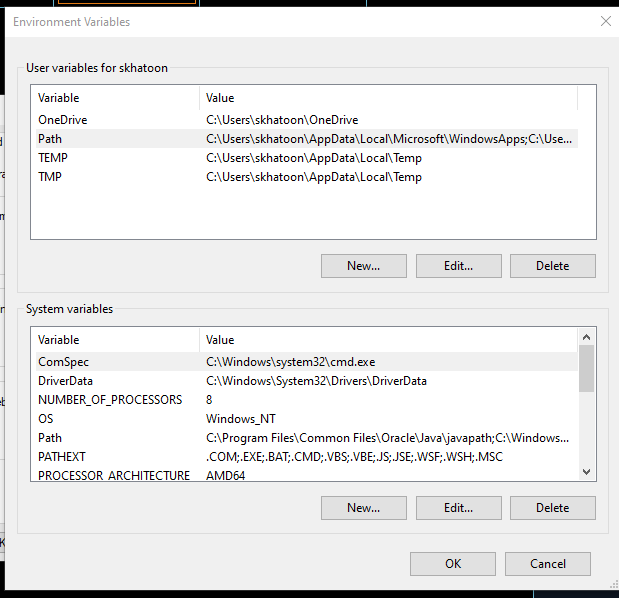




After downloading zip file unzip the folder and copy the folder in C Drive. Now open the folder and open bin folder and copy the path of bin folder. Now click on start search for environment. Open system environment variables.



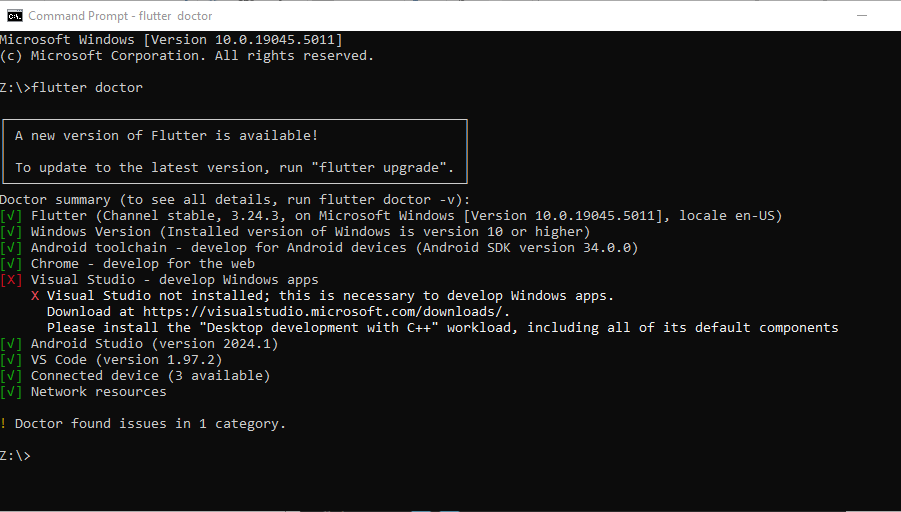
Now go to the environment variables setting.



Now double click on path and add new path and paste be bin file path. Press ok.

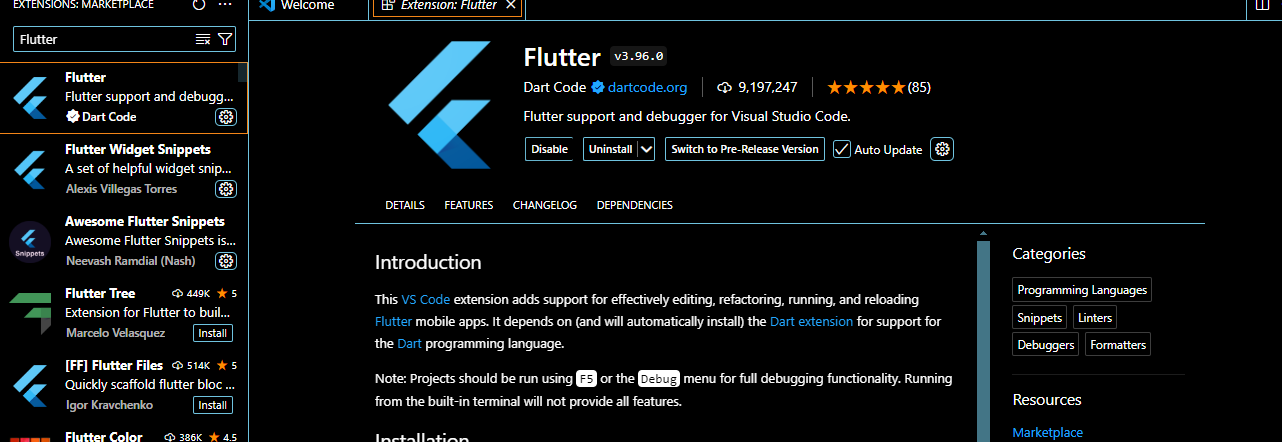
**Check Installation Progress**

After that open cmd and write flutter in command prompt. In CMD run command “flutter doctor” to check the installation is successful or not.

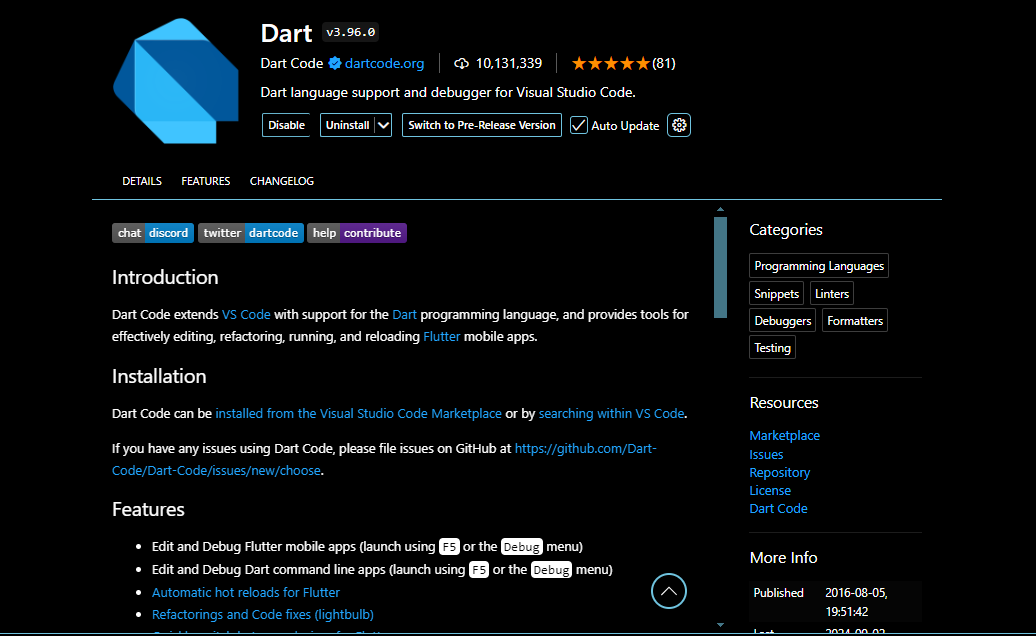


**Extension Installation**

Now install extension of flutter in VS code

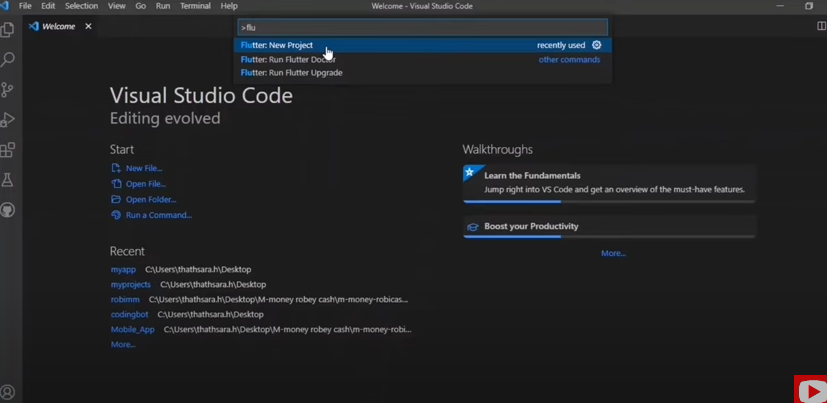


Now install dart extension.

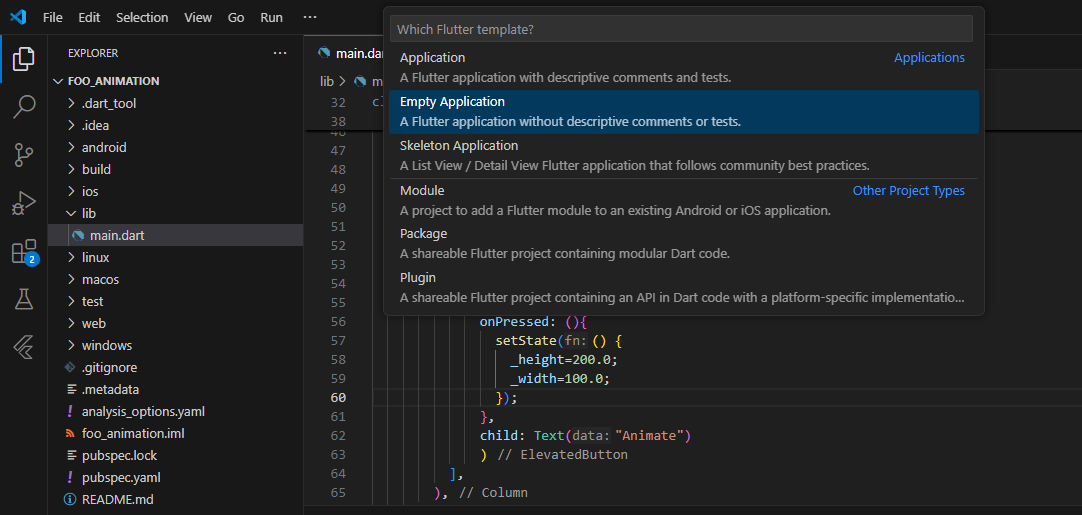


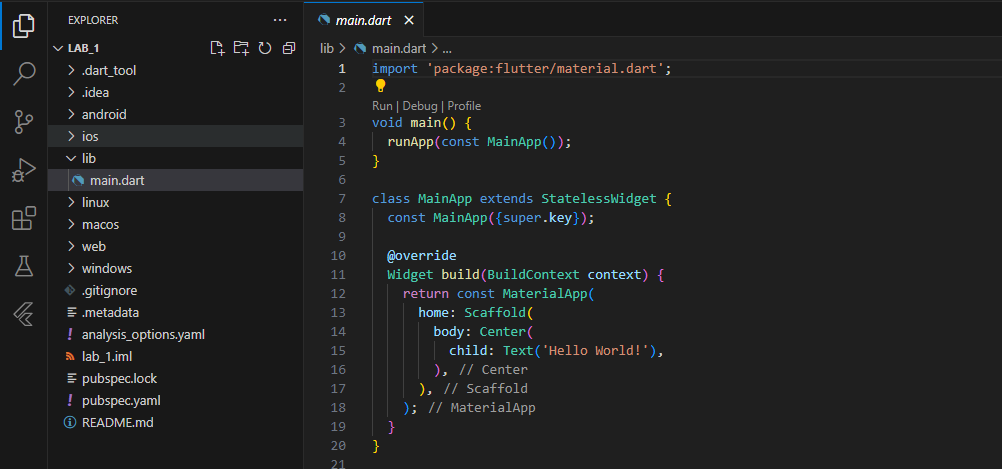
**Create First Application on Flutter**

Go view option and open command palette and write flutter select new file option also what you want to create.



Select empty application from the drop down.





Now run this app.

**Exploring project structure and main files**

.**idea:**

The. idea folder contains your code editors project related settings specific files. No need to make any manual changes here.

**Android:**

In this folder, if you want to do any code specific to android. The android folder inside the flutter project contains the android platform specific settings, resources and code. If you need to write any platform specific code, you may be making changes in here.

**Build:**

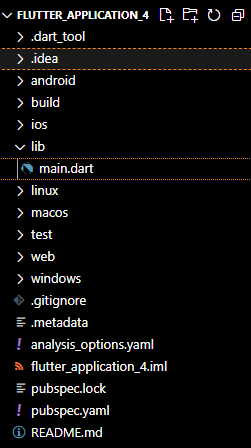
The build folder contains the output generated when build or run the flutter project. You will find all the release files/folders like apks and app bundles here.

**iOS:**

Similar to android folder the iOS, the iOS folder contains iOS specific settings, resources and code. You will make the platform specific code or changes in here.

**Lib:**

The lib folder is main folder where you will write all the flutter app related codes. Initially it will contain just one single file main. Art which has the entry point for flutter app.



**Test:**

The test folder is set to contain any testing related codes that you write. If you decide to write tests/test cases, you will be adding code in here.

**.gitignore:**

The next file is .gitignore. This is git specific file. You can adjust the .gitignore file to include/exclude ant files/folder as your need.

**Metadata:**

As name suggest, the. metadata files contain Flutter project related metadata that tools use. You should not make any manual changes to this file.

**.pubspec.lock:**

The pubspec.lock file is helper file create next to pubspec.yaml file. It list the specific versions of each dependency that packages use in your app and ensure the version stays consistent across different developer machine. No need to take any changes here.

**.pubspec.yaml:**

The pubspec.yaml contains flutter app specific metadata and configurations. You can configure dependencies such as external packages, image assets, font files, app versions etc. with help of this file. You will often make changes to the pubspec.yaml file add external dependencies.

**README.md:**

The readme.md is markdown format file which is primarily used to describe your project in git repository. You can write project specific thins like what your project does or how to use certain libraries in the README.md file.

**Basic widgets:**

Flutter comes with a suite of powerful basic widgets, of which the following are commonly used:

**Text**

The Text widget lets you create a run of styled text within your application.

**Row, Column**

These flex widgets let you create flexible layouts in both the horizontal (Row) and vertical (Column) directions. The design of these objects is based on the web's flexbox layout model.

**Stack**

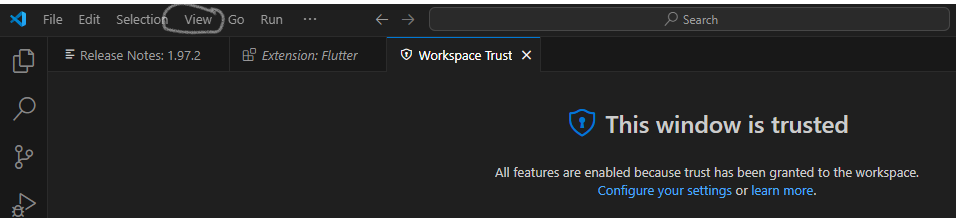
Instead of being linearly oriented (either horizontally or vertically), a Stack widget lets you place widgets on top of each other in paint order. You can then use the Positioned widget on children of a Stack to position them relative to the top, right, bottom, or left edge of the stack. Stacks are based on the web's absolute positioning layout model.

**Container**

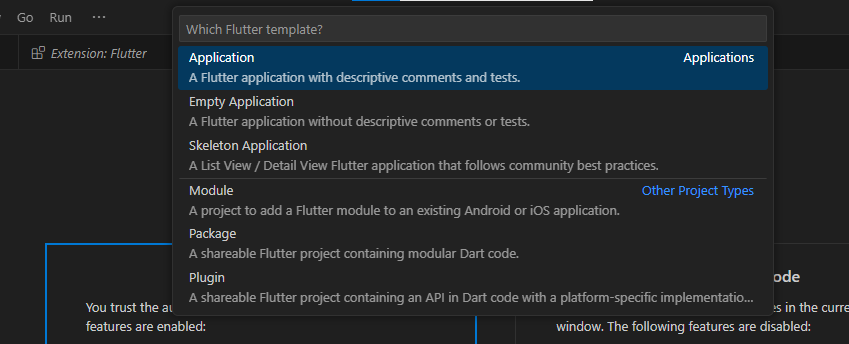
The Container widget lets you create a rectangular visual element. A container can be decorated with a Box Decoration, such as a background, a border, or a shadow. A Container can also have margins, padding, and constraints applied to its size. In addition, a Container can be transformed in three-dimensional space using a matrix.

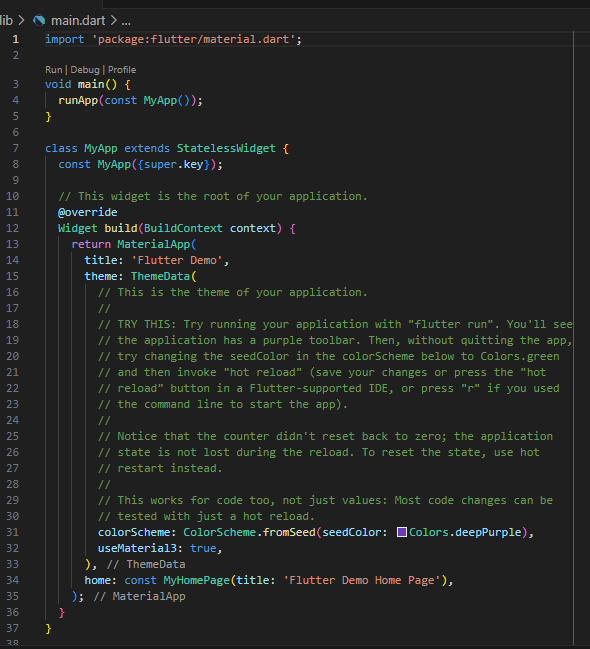
**Build-in Counter App:**

Click on the view button from task panel. Then select command palette. There also short key ctrl+shift+P to command palette.

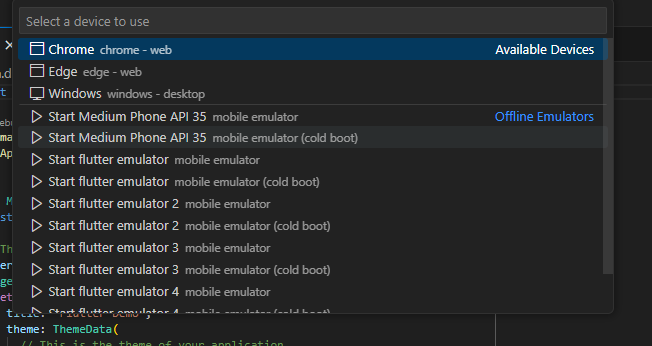


After that then select project for palate. After that select template of project. Now we select application. This application contain code od basic counter app.

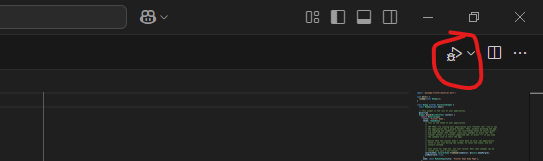




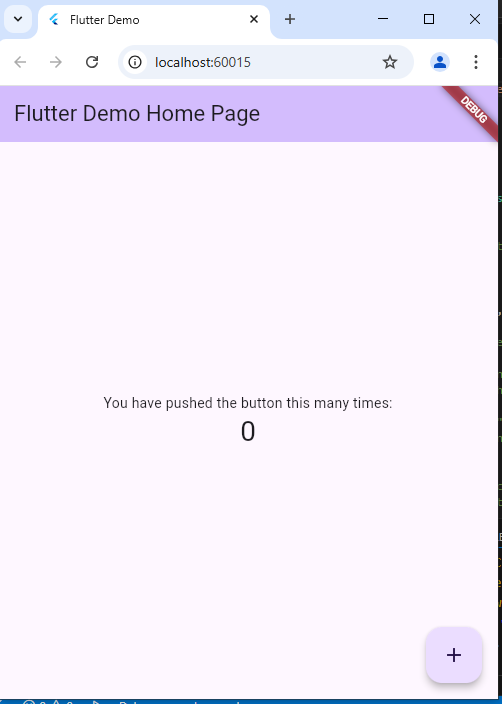
To run this code, we have to select device. We select chrome browser device.



Now run you code.



**Output**



**Assessments:**

1. What is the role of the Flutter extension in Visual Studio Code, and how does it enhance the development experience for Flutter applications?
2. How does the debugging functionality of the Flutter extension in Visual Studio Code improve the efficiency of troubleshooting Flutter applications?
3. Create a simple Flutter app that displays "Your introduction and your hobbies" on the screen. Submit screen shot of output.
4. Create counter application that we created in lab. Your task is adding another button for decrement the counter values. Add floating button for decrement.