Name : Arslan Tahir

Sap id : 47120

Section : Bscs (5-2) Mad

Topic : Assignment no 1 OF MAD

Submitted To : Sir Waqar Arshad

Date Of Submission : Thursday sept 18th 2025

Task 1: Flutter State Management (Bidding Page) Design and implement a bidding page in Flutter.

- The page should display the user's current maximum bid on a product.
- When the ***"Increase Bid"*** button is tapped, the bid amount should increase by $50.
- To achieve this, create a ***MaximumBid*** class that extends the ***StatefulWidget*** class, and a ***_MaximumBidState*** class that manages the state and handles the bid increment functionality.

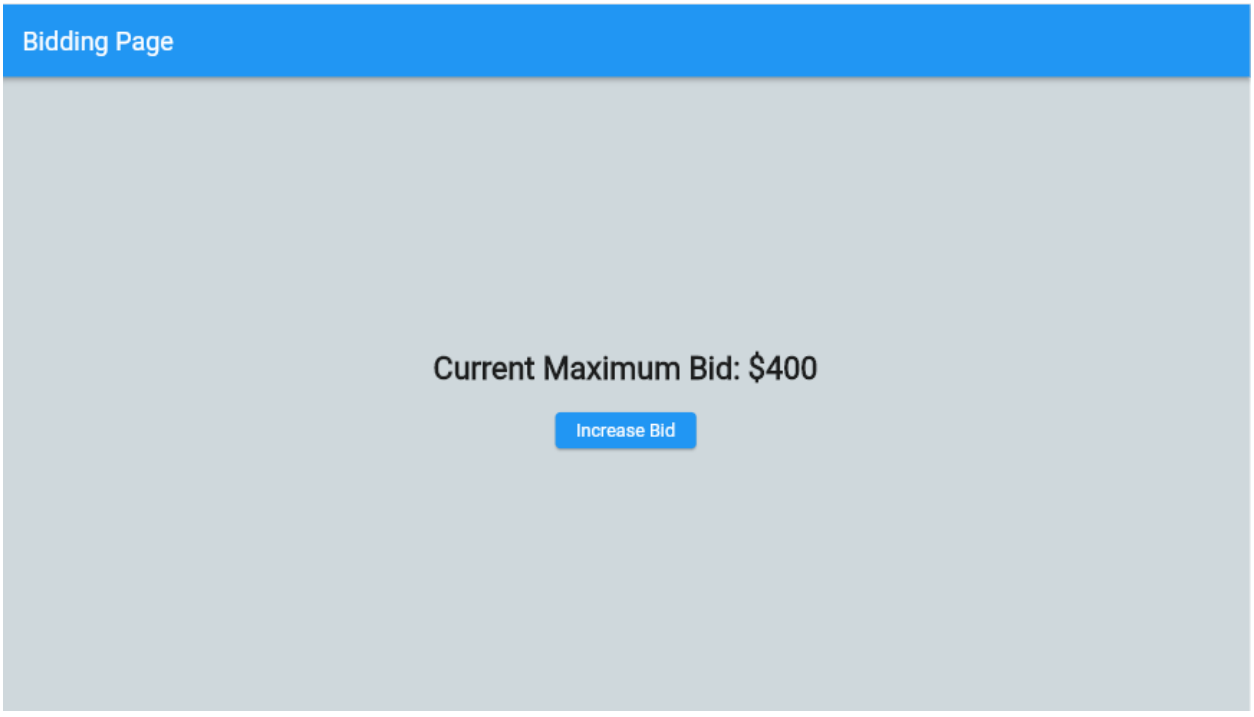
ASSINGMENT STARTS BELOW

CODE OF TASK NO 1 :

```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}
class MyApp extends StatelessWidget {   const MyApp({Key? key}) :
super(key: key);

  @override
  Widget build(BuildContext context) {     return MaterialApp(
      debugShowCheckedModeBanner: false,       home: const BiddingPage(),
    );
  }
}
class BiddingPage extends StatelessWidget {   const BiddingPage({Key? key}) :
super(key: key);

  @override
  Widget build(BuildContext context) {     return Scaffold(
      appBar: AppBar(title: const Text('Bidding Page')),       body: const Center(         child:
MaximumBid(),
      ),
      backgroundColor: Colors.blueGrey[100], // Set background color
    );
  }
}
class MaximumBid extends StatefulWidget {
  const MaximumBid({Key? key}) : super(key: key);
  @override
  _MaximumBidState createState() => _MaximumBidState();
}
class _MaximumBidState extends State<MaximumBid> {   int _currentBid = 100;
```

```
 void _increaseBid() => setState(() => _currentBid += 50);
 @override
 Widget build(BuildContext context) {     return Column(
    mainAxisAlignment: MainAxisAlignment.center,       children: [
      Text(
        'Current Maximum Bid: \$$_currentBid',
        style: const TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
      ),
      const SizedBox(height: 20),         ElevatedButton(
        onPressed: _increaseBid,
        child: const Text('Increase Bid'),
      ),
    ],
  );
 }
}
```

Screen Shots of the code Outputs .



Task 2: Dart Basics (Input, Loops, Conditionals & Lists)

Write a Dart console program that:

1. Takes input from the user for their name and age.

o If the age is less than 18, print: "Sorry [Name], you are not eligible to register." and stop execution.
o Otherwise, continue.

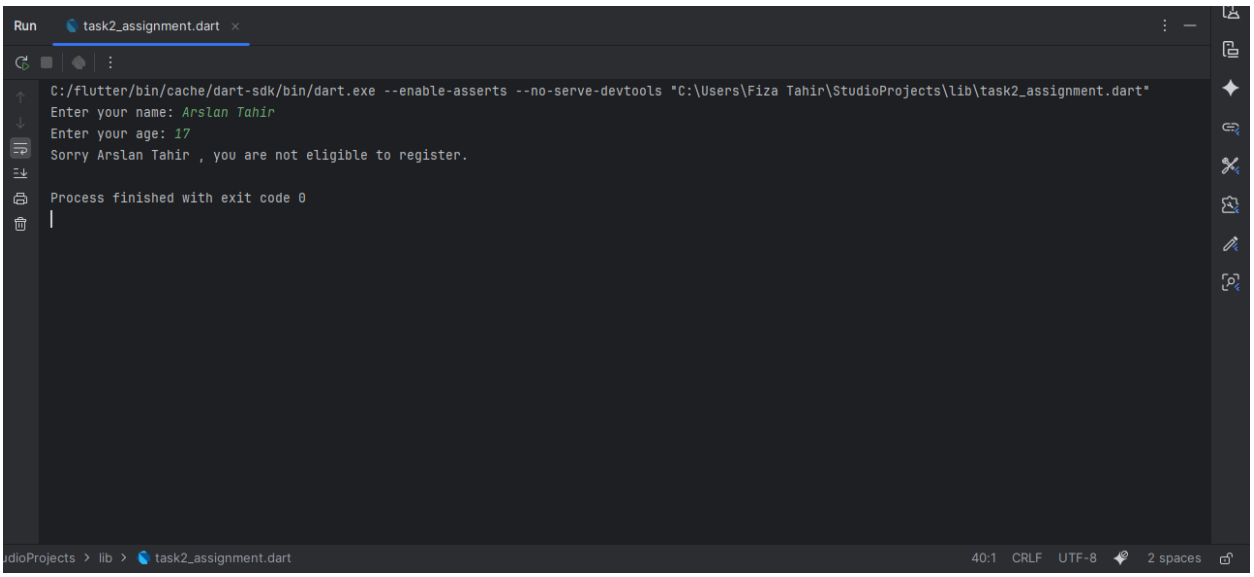
2. Ask the user to enter N numbers (the program should first ask how many numbers the user wants to enter).
3. Store all numbers in a list and calculate:
   o The sum of even numbers.
   o The sum of odd numbers.
   o The largest number entered.
   o The smallest number entered.
4. Print the results clearly

```dart
import 'dart:io';

void main() {
  // Step 1: Take input
  stdout.write("Enter your name: ");
  String name = stdin.readLineSync()!;

  stdout.write("Enter your age: ");
  int age = int.parse(stdin.readLineSync()!);

  if (age < 18) {
    print("Sorry $name, you are not eligible to register.");
    return; // stop execution
  }

  // Step 2: Ask how many numbers
  stdout.write("How many numbers do you want to enter? ");
  int n = int.parse(stdin.readLineSync()!);

  List<int> numbers = [];

  for (int i = 0; i < n; i++) {
    stdout.write("Enter number ${i + 1}: ");
    numbers.add(int.parse(stdin.readLineSync()!));
  }

  // Step 3: Calculate sums
  int sumEven = numbers.where((x) => x % 2 == 0).fold(0, (a, b) => a + b);
  int sumOdd = numbers.where((x) => x % 2 != 0).fold(0, (a, b) => a + b);
  int largest = numbers.reduce((a, b) => a > b ? a : b);
  int smallest = numbers.reduce((a, b) => a < b ? a : b);

  // Step 4: Print results
  print("\nResults:");
  print("Sum of even numbers: $sumEven");
  print("Sum of odd numbers: $sumOdd");
```

```
  print("Largest number: $largest");
  print("Smallest number: $smallest");
}
```

Screen shot of code 2 _ if age is greater than 18 .

Screen shot of code 2 _ if age is not greater than 18 .



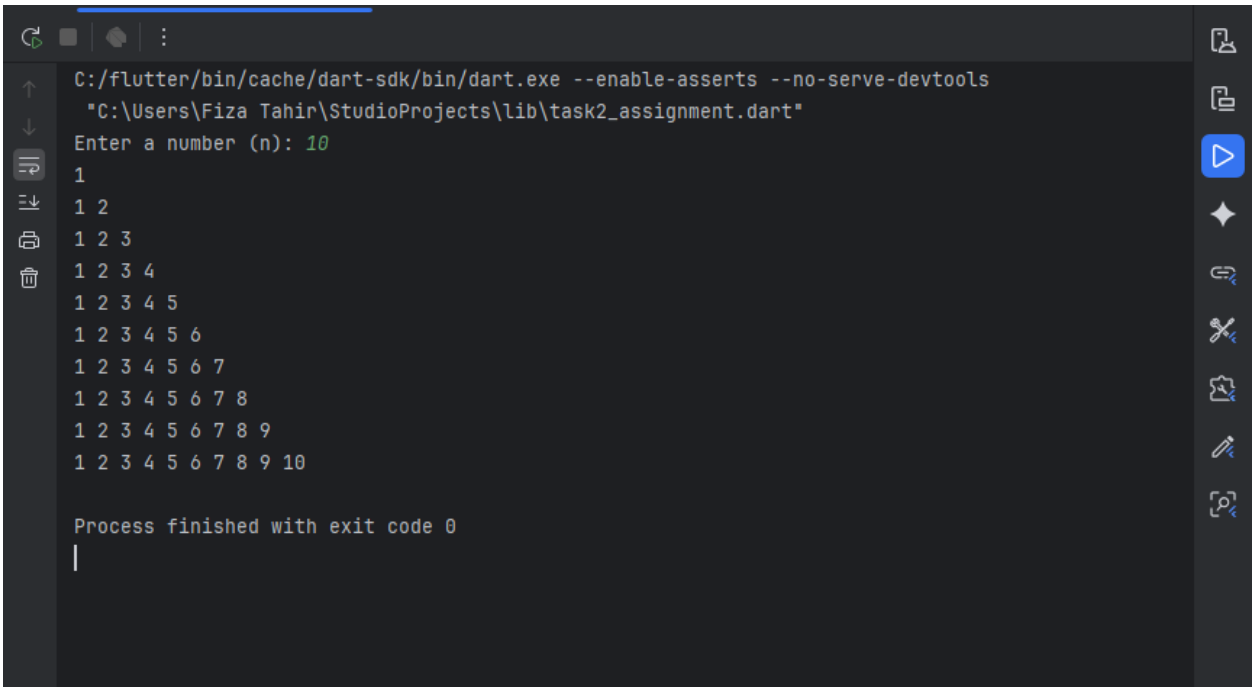
Task 3: Dart Loops & Patterns

Write a program in Dart that takes an integer n from the user and prints the following number pyramid pattern using nested loops: Sample Input: n = 5 Sample Output:

Code OF Task 3

```dart
import 'dart:io';

void main() {
  // Ask the user for input
  stdout.write("Enter a number (n): ");
  int n = int.parse(stdin.readLineSync()!);

  // Print the pyramid pattern using nested loops
  for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= i; j++) {
      stdout.write("$j ");
    }
    print(""); // Move to next line after each row
  }
}
```

Screen Shot of output :



End of the assignment no 1 (OF MAD) .