ISS STOXX

Assignment-1

Aryaman Agarwal

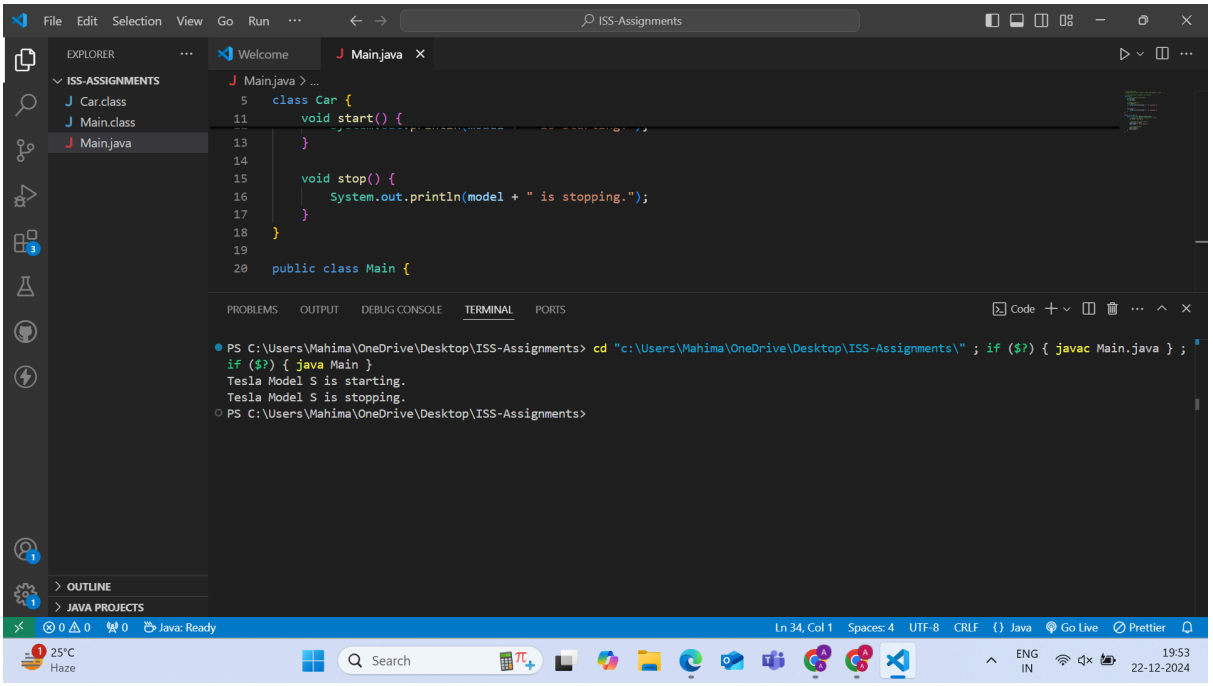
EmployeeID: 16333

1. Core Java

1.1- Basics of Java– Class, object
Code:

```
// File: Main.java
// Demonstrating Class, Object, State, and Methods in Java

// Class definition (blueprint for objects)
class Car {
    // State (Instance Variables)
    String color;
    String model;

    // Method (Behavior)
    void start() {
        System.out.println(model + " is starting.");
    }

    void stop() {
        System.out.println(model + " is stopping.");
    }
}

public class Main {
    public static void main(String[] args) {
        // Creating an object (Instance of a class)
        Car myCar = new Car();

        // Setting state (attributes)
        myCar.color = "Red";
        myCar.model = "Tesla Model S";

        // Invoking methods
        myCar.start();
        myCar.stop();
```

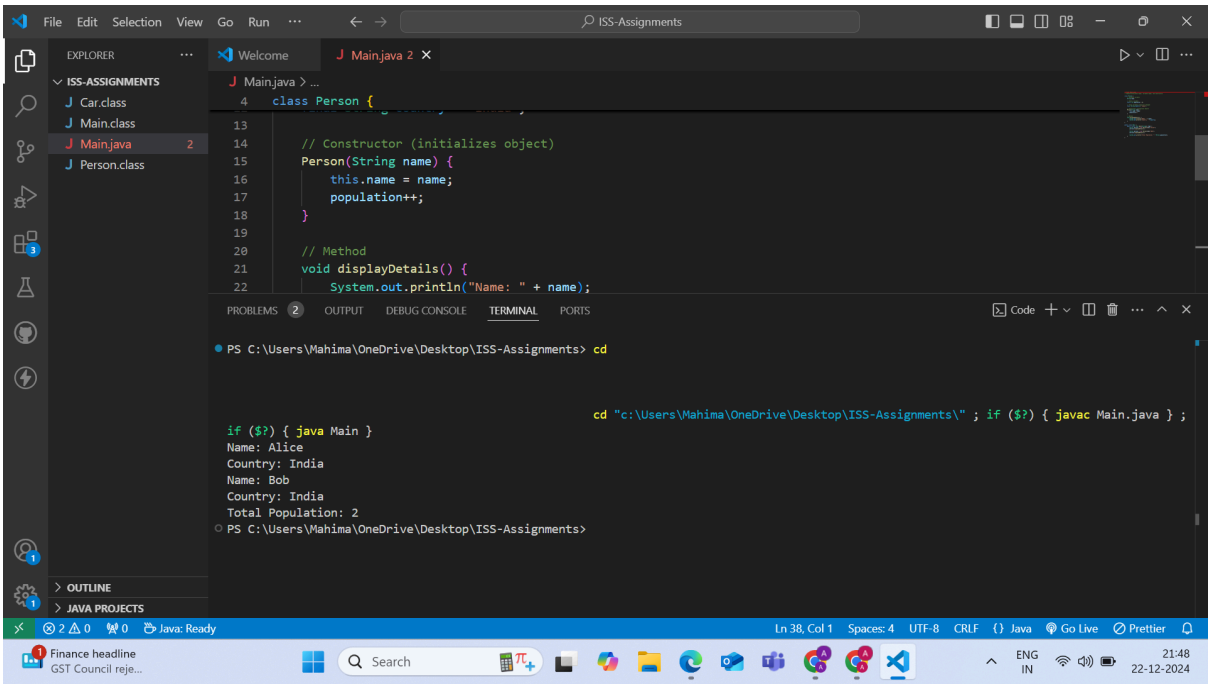
```
    }
}
```



Code:

```
// File: Main.java
// Demonstrating data types, variable types, and constructors

class Person {
    // Instance variable
    String name;

    // Static variable
    static int population = 0;

    // Final variable (cannot be changed)
    final String country = "India";

    // Constructor (initializes object)
    Person(String name) {
        this.name = name;
        population++;
    }
```

```
    // Method
    void displayDetails() {
        System.out.println("Name: " + name);
        System.out.println("Country: " + country);
    }
}

public class Main {
    public static void main(String[] args) {
        Person person1 = new Person("Alice");
        person1.displayDetails();

        Person person2 = new Person("Bob");
        person2.displayDetails();

        System.out.println("Total Population: " + Person.population);
    }
}
```



Code: Loop control and decision making

```
// File: Main.java
// Demonstrating Loops and Decision Making
```

```
public class Main {
    public static void main(String[] args) {
        // Print numbers from 1 to 5 using a loop
        for (int i = 1; i <= 5; i++) {
            System.out.println("Number: " + i);
        }

        // Check if a number is even or odd
        int num = 4;
        if (num % 2 == 0) {
            System.out.println(num + " is even.");
        } else {
            System.out.println(num + " is odd.");
        }
    }
}
```

Code: Strings and arrays

```
// Demonstrating Strings and Arrays

public class Main {
    public static void main(String[] args) {
        // String example
        String message = "Hello, Java!";
        System.out.println("Message: " + message);
        System.out.println("Length: " + message.length());

        // Array example
        int[] numbers = {5, 10, 15, 20};
        System.out.println("First number: " + numbers[0]);

        // Sort and print array
        java.util.Arrays.sort(numbers);
        for (int num : numbers) {
            System.out.println(num);
        }
    }
}
```

Code: Inheritance, Polymorphism, Abstraction, Encapsulation, Interfaces, Packages

```
// File: Main.java
```

```
// Demonstrating Object-Oriented Concepts

// Base Class (Parent)
class Animal {
    void sound() {
        System.out.println("Some generic animal sound");
    }
}

// Derived Class (Child) - Inheritance
class Dog extends Animal {
    // Overriding
    @Override
    void sound() {
        System.out.println("Bark");
    }
}

// Interface
interface Pet {
    void play();
}

// Class implementing interface
class Cat implements Pet {
    public void play() {
        System.out.println("Cat is playing.");
    }
}

// Abstract Class
abstract class Vehicle {
    abstract void start();
}

class Car extends Vehicle {
    void start() {
        System.out.println("Car is starting.");
    }
}

public class Main {
    public static void main(String[] args) {
```

```
        // Inheritance and Overriding
        Animal myDog = new Dog();
        myDog.sound();

        // Interface
        Pet myCat = new Cat();
        myCat.play();

        // Abstraction
        Vehicle myCar = new Car();
        myCar.start();
    }
}
```

Code: Exception handling

```
// File: Main.java
// Demonstrating Exception Handling

import java.io.*;

public class Main {
    // Method using 'throws' to declare exceptions
    public static void readFile(String filePath) throws IOException {
        BufferedReader br = new BufferedReader(new FileReader(filePath));
        System.out.println(br.readLine());
        br.close();
    }

    public static void main(String[] args) {
        try {
            // Demonstrating 'throw'
            readFile("nonexistentfile.txt");
        } catch (IOException e) {
            System.out.println("File not found: " + e.getMessage());
        } finally {
            System.out.println("Execution completed.");
        }
    }
}
```

Code: List, set and map

```
// File: Main.java
// Demonstrating List, Set, Map with custom object Student

import java.util.*;

// Custom Object
class Student {
    int id;
    String name;

    Student(int id, String name) {
        this.id = id;
        this.name = name;
    }

    @Override
    public String toString() {
        return "ID: " + id + ", Name: " + name;
    }
}

public class Main {
    public static void main(String[] args) {
        // List
        List<Student> studentList = new ArrayList<>();
        studentList.add(new Student(1, "Alice"));
        studentList.add(new Student(2, "Bob"));
        System.out.println("List: " + studentList);

        // Set
        Set<Student> studentSet = new HashSet<>(studentList);
        System.out.println("Set: " + studentSet);

        // Map
        Map<Integer, Student> studentMap = new HashMap<>();
        for (Student s : studentList) {
            studentMap.put(s.id, s);
        }
        System.out.println("Map: " + studentMap);
    }
}
```

Code: Sorting

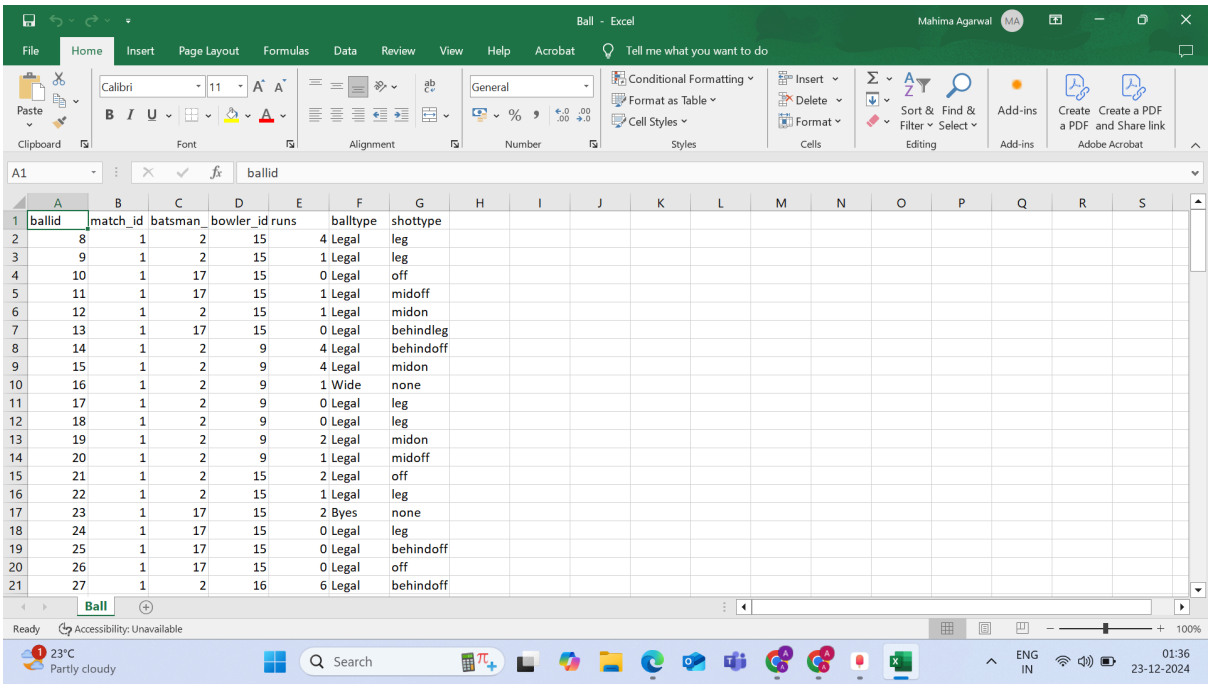
```
// File: Main.java
// Demonstrating Sorting using Comparable and Comparator

import java.util.*;

class Student implements Comparable<Student> {
    int id;
    String name;

    Student(int id, String name) {
        this.id = id;
        this.name = name;
    }

    @Override
    public int compareTo(Student o) {
        return this.id - o.id; // Natural sorting by ID
    }

    @Override
    public String toString() {
        return "ID: " + id + ", Name: " + name;
    }
}

class SortByName implements Comparator<Student> {
    @Override
    public int compare(Student s1, Student s2) {
        return s1.name.compareTo(s2.name); // Custom sorting by Name
    }
}

public class Main {
    public static void main(String[] args) {
        List<Student> students = new ArrayList<>();
        students.add(new Student(2, "Bob"));
        students.add(new Student(1, "Alice"));

        // Sort by ID (Natural Order)
        Collections.sort(students);
        System.out.println("Sorted by ID: " + students);
```

```
        // Sort by Name (Custom Order)
        Collections.sort(students, new SortByName());
        System.out.println("Sorted by Name: " + students);
    }
}
```

Code: Multithreading

```
// File: Main.java
// Demonstrating Multithreading

import java.util.concurrent.*;

class MyTask implements Runnable {
    private final int taskId;

    MyTask(int taskId) {
        this.taskId = taskId;
    }

    @Override
    public void run() {
        System.out.println("Executing Task ID: " + taskId + " by " +
Thread.currentThread().getName());
    }
}

public class Main {
    public static void main(String[] args) {
        // Thread Pool Executor
        ExecutorService executor = Executors.newFixedThreadPool(2);

        for (int i = 1; i <= 5; i++) {
            executor.execute(new MyTask(i));
        }

        executor.shutdown();
    }
}
```

Database creation and management:



ballid	match_id	batsman_	bowler_id	runs	balltype	shottype
8	1	2	15	4	Legal	leg
9	1	2	15	1	Legal	leg
10	1	17	15	0	Legal	off
11	1	17	15	1	Legal	midoff
12	1	2	15	1	Legal	midon
13	1	17	15	0	Legal	behindleg
14	1	2	9	4	Legal	behindoff
15	1	2	9	4	Legal	midon
16	1	2	9	1	Wide	none
17	1	2	9	0	Legal	leg
18	1	2	9	0	Legal	leg
19	1	2	9	2	Legal	midon
20	1	2	9	1	Legal	midoff
21	1	2	15	2	Legal	off
22	1	2	15	1	Legal	leg
23	1	17	15	2	Byes	none
24	1	17	15	0	Legal	leg
25	1	17	15	0	Legal	behindoff
26	1	17	15	0	Legal	off
27	1	2	16	6	Legal	behindoff

```
create table Matchh(
    match_id int AUTO_INCREMENT primary key,
    man_of_match varchar(20),
    team_id1 int ,
    team_id2 int ,
    match_date date ,
    scorecard_id int
);

create table Team(
    team_id int auto_increment primary key,
    captain varchar(25),
    ranking int,
    coach varchar(25),
    country varchar (25),
    no_losses int DEFAULT(0),
    no_wins int DEFAULT(0)
);

create table Player(
    player_id int auto_increment primary key,
    player_name varchar(25),
    runs int DEFAULT(0),
    wickets int DEFAULT(0),
    team_id INT,
```

```
    FOREIGN KEY (team_id) REFERENCES Team(team_id),
    player_type enum('Batsman','Bowler','Allrounder')

);

create table Scorecard(
    scorecard_id int auto_increment primary key,
    highest_run varchar(25),
    highest_wicket varchar(25),
    match_id int,
    foreign key (match_id) references Matchh(match_id),
    winner int,
    foreign key (winner) references Team(team_id)
);

alter table Matchh
add foreign key (team_id1)
references Team(team_id)
on delete set null;

alter table Matchh
add foreign key (team_id2)
references Team(team_id)
on delete set null;

alter table Matchh
add foreign key (scorecard_id)
references Scorecard(scorecard_id)
on delete set null;

insert into Team values (1,'Rohit Sharma',1,'Rahul Dravid','India',22,51);
insert into Team values (2,'Jos Butler',3,'Nasser Hussain','England',32,47);
insert into Team values (3,'Babar Azam',5,'Gary Kirsten','Pakistan',62,22);
insert into Team values (4,'Pat Cummins',2,'Justin Langer','Australia',42,55);
insert into Team values (5,'Rovman Powell',4,'Gary Sobers','West Indies',27,31);

select * from Team;

insert into Player values (1,'Rohit Sharma',4509,10,1,'Batsman');
insert into Player values (2,'Virat Kohli',5109,21,1,'Batsman');
insert into Player values (3,'Babar Azam',3511,7,3,'Batsman');
insert into Player values (4,'Shaheen Afridi',670,231,3,'Bowler');
insert into Player values (5,'Pat Cummins',867,257,4,'Bowler');
```

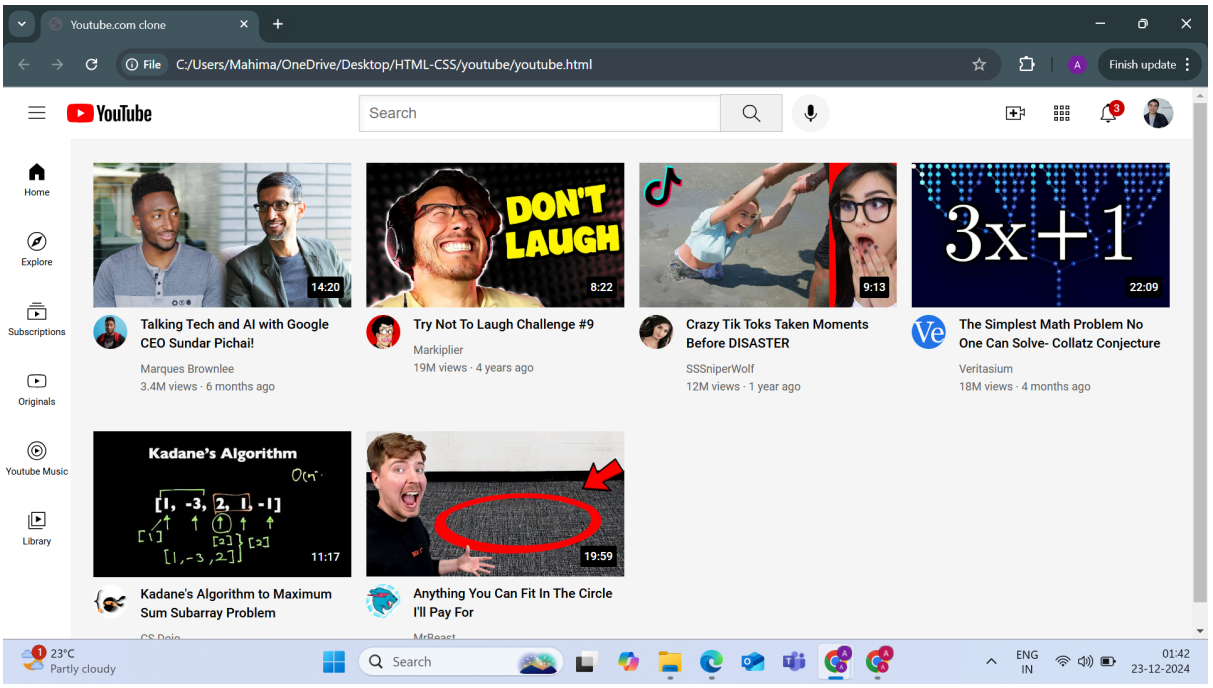
```
insert into Player values (6,'David Warner',3608,5,4,'Batsman');
insert into Player values (7,'Andre Russell',2203,114,5,'Allrounder');
insert into Player values (8,'Rovman Powell',1772,89,5,'Allrounder');
insert into Player values (9,'Sam Curran',1267,157,2,'Bowler');
insert into Player values (10,'Moen Ali',3003,104,2,'Allrounder');
insert into Player values (11,'Jos Butler',4028,0,2,'Batsman');
insert into Player (player_name, runs, wickets, team_id, player_type) values ('Jasprit
Bumrah',112,402,1,'Bowler');


insert into Player values (13,'Mark Wood',197,166,2,'Bowler');
insert into Player values (14,'Liam Livingstone',3212,56,2,'Allrounder');
insert into Player values (15,'Jofra Archer',86,102,2,'Bowler');
insert into Player values (16,'Reece Topley',288,123,2,'Bowler');

insert into Player values (17,'Rishab Pant',2203,0,1,'Batsman');
insert into Player values (18,'Suryakumar Yadav',3401,8,1,'Batsman');
insert into Player values (19,'Shubman Gill',1576,5,1,'Batsman');
insert into Player values (20,'Hardik Pandya',3000,97,1,'Allrounder');
insert into Player values (21,'Ravindra Jadeja',2111,134,1,'Allrounder');

select * from Player;

insert into Matchh values (1,'Virat Kohli',1,2,'2016-09-07',null);
insert into Matchh values (2,'Babar Azam',3,4,'2018-10-22',null);
insert into Matchh values (3,'Rohit Sharma',1,5,'2011-02-13',null);
insert into Matchh values (4,'David Warner',4,2,'2015-09-07',null);
insert into Matchh values (5,'Virat Kohli',1,3,'2014-04-15',null);
insert into Matchh values (6,'Andre Russell',4,5,'2010-04-29',null);
insert into Matchh values (7,'Jos Butler',3,2,'2013-12-05',null);
insert into Matchh values (8,'Pat Cummins',1,4,'2019-11-11',null);
insert into Matchh values (9,'Moen Ali',5,2,'2020-12-01',null);
insert into Matchh values (10,'Rovman Powell',3,5,'2023-11-19',null);


select * from Matchh;

insert into Scorecard values (101,'Virat Kohli','Sam Curran',1,1);
insert into Scorecard values (102,'Babar Azam','Moen Ali',2,3);
insert into Scorecard values (103,'Rohit Sharma','Shaheen Afridi',3,1);
insert into Scorecard values (104,'David Warner','Pat Cummins',4,4);
insert into Scorecard values (105,'Virat Kohli','Shaheen Afridi',5,1);
insert into Scorecard values (106,'Rovman Powell','Andre Russell',6,5);
```

```
insert into Scorecard values (107,'Jos Butler','Shaheen Afridi',7,2);
insert into Scorecard values (108,'Rohit Sharma','Pat Cummins',8,4);
insert into Scorecard values (109,'Jos Butler','Moen Ali',9,2);
insert into Scorecard values (110,'Babar Azam','Rovman Powell',10,5);

select * from Scorecard;

update Matchh set scorecard_id = 101 where match_id = 1;
update Matchh set scorecard_id = 102 where match_id = 2;
update Matchh set scorecard_id = 103 where match_id = 3;
update Matchh set scorecard_id = 104 where match_id = 4;
update Matchh set scorecard_id = 105 where match_id = 5;
update Matchh set scorecard_id = 106 where match_id = 6;
update Matchh set scorecard_id = 107 where match_id = 7;
update Matchh set scorecard_id = 108 where match_id = 8;
update Matchh set scorecard_id = 109 where match_id = 9;
update Matchh set scorecard_id = 110 where match_id = 10;

create table Ball(
    ballid int auto_increment primary key,
    match_id int DEFAULT(1),
    foreign key (match_id) references Matchh(match_id),
    batsman_id int ,
    foreign key (batsman_id) references Player (player_id),
    bowler_id int,
    foreign key (bowler_id) references Player (player_id),
    runs int ,
    balltype enum('Legal','Wicket','Wide','Noball','Legbye','Byes'),
    shottype enum('midoff','midon','off','leg','behindoff','behindleg','none')

);

/* For adding 2nd match alter the Ball table and set match id default to 2 etc */
insert into Ball values (1,1,2,9,4,'Legal','off');
insert into Ball (batsman_id,bowler_id, runs, balltype,shottype) values (2,9,0,'Legal','midoff');
insert into Ball (batsman_id,bowler_id, runs, balltype,shottype) values (2,9,1,'Legal','leg');
insert into Ball (batsman_id,bowler_id, runs, balltype,shottype) values
(1,9,0,'Legal','behindleg');
insert into Ball (batsman_id,bowler_id, runs, balltype,shottype) values (1,9,6,'Legal','leg');
insert into Ball (batsman_id,bowler_id, runs, balltype,shottype) values (1,9,1,'Wide','none');
insert into Ball (batsman_id,bowler_id, runs, balltype,shottype) values (1,9,0,'Wicket','none');
select * from Ball;
```

select sum(runs),bowler_id from Ball group by bowler_id;

select player_id, player_name from Player where team_id = 1 OR team_id = 2;

HTML and CSS




```
<!DOCTYPE html>
<html>
    <head>
        <title>Youtube.com clone</title>
        <link rel="preconnect" href="https://fonts.googleapis.com">
        <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
        <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500;700&display=swap"
rel="stylesheet">

        <link rel="stylesheet" href="styles/general.css">
        <link rel="stylesheet" href="styles/video.css">
        <link rel="stylesheet" href="styles/headers.css">
        <link rel="stylesheet" href="styles/sidebar.css">


    </head>

    <body>
```

```
<header class="header">
  <div class="left-section">
    <img class="hamburger-menu" src="icons/hamburger-menu.svg">
    <img class="youtube-logo" src="icons/youtube-logo.svg">
  </div>

  <div class="middle-section">
    <input class="search-bar" type="text" placeholder="Search">
    <button class="search-button">
      <img class="search-icon" src="icons/search.svg">
      <div class="tooltip">Search</div>
    </button>
    <button class="voice-search-button">
      <img class="voice-search-icon" src="icons/voice-search-icon.svg">
      <div class="tooltip">Search with your voice</div>
    </button>
  </div>

  <div class="right-section">
    <div class="upload-icon-container">
      <img class="upload-icon" src="icons/upload.svg">
      <div class="tooltip">Create</div>
    </div>

    <div class="youtube-apps-icon-container">
      <img class="youtube-apps-icon" src="icons/youtube-apps.svg">
      <div class="tooltip">YouTube Apps</div>
    </div>

    <div class="notifications-icon-container">
      <img class="notifications-icon" src="icons/notifications.svg">
      <div class="notifications-count">3</div>
      <div class="tooltip">Notifications</div>
    </div>

    <img class="current-user-picture" src="channelpics/my-channel.jpeg">
  </div>

</header>

<nav class="sidebar">
  <div class="sidebar-link">
    <img src="icons/home.svg">
```

```
      <div>Home</div>
    </div>

    <div class="sidebar-link">
      <img src="icons/explore.svg">
      <div>Explore</div>
    </div>

    <div class="sidebar-link">
      <img src="icons/subscriptions.svg">
      <div>Subscriptions</div>
    </div>

    <div class="sidebar-link">
      <img src="icons/originals.svg">
      <div>Originals</div>
    </div>

    <div class="sidebar-link">
      <img src="icons/youtube-music.svg">
      <div>Youtube Music</div>
    </div>

    <div class="sidebar-link">
      <img src="icons/library.svg">
      <div>Library</div>
    </div>
  </nav>
  <!-- This is a comment-->
  <div class="video-grid">
    <div class="video-preview">

      <div class="thumbnail-row">
        <a href="https://www.youtube.com/watch?v=n2RNcPRtAiY">
          <img class="thumbnail" src="thumbnails/thumbnail-1.webp">
        </a>

        <div class="video-time">14:20</div>
      </div>

      <div class="video-info-grid">
        <div class="channel-picture">
          <img class="profile-picture" src="channelpics/channel-1.jpeg">
```

```
        </div>

        <div class="video-info">
          <p class="video-title">
            <a href="https://www.youtube.com/watch?v=n2RNcPRtAiY">
            Talking Tech and AI with Google CEO Sundar Pichai!
            </a>
          </p>
          <p class="video-author">
            Marques Brownlee
          </p>
          <p class="video-stats">
            3.4M views &#183; 6 months ago
          </p>
        </div>


    </div>
</div>

<div class="video-preview">

    <div class="thumbnail-row">
        <img class="thumbnail" src="thumbnails/thumbnail-2.webp">
        <div class="video-time">8:22</div>
    </div>

    <div class="video-info-grid">
        <div class="channel-picture">
            <img class="profile-picture" src="channelpics/channel-2.jpeg">
        </div>

        <div class="video-info">
          <p class="video-title">
            <a href="https://www.youtube.com/watch?v=mP0RAo9SKZk">
              Try Not To Laugh Challenge #9
            </a>

          </p>
          <p class="video-author">
            Markiplier
          </p>
          <p class="video-stats">
```

```
            19M views &#183; 4 years ago
          </p>
        </div>


    </div>
</div>

<div class="video-preview">

    <div class="thumbnail-row">
        <img class="thumbnail" src="thumbnails/thumbnail-3.webp">
        <div class="video-time">9:13</div>
    </div>

    <div class="video-info-grid">
        <div class="channel-picture">
            <img class="profile-picture" src="channelpics/channel-3.jpeg">
        </div>

        <div class="video-info">
            <p class="video-title">
                Crazy Tik Toks Taken Moments Before DISASTER
            </p>
            <p class="video-author">
                SSSniperWolf
            </p>
            <p class="video-stats">
                12M views &#183; 1 year ago
            </p>
        </div>


    </div>
</div>

<div class="video-preview">

    <div class="thumbnail-row">
        <img class="thumbnail" src="thumbnails/thumbnail-4.webp">
        <div class="video-time">22:09</div>
    </div>
```

```
    <div class="video-info-grid">
      <div class="channel-picture">
        <img class="profile-picture" src="channelpics/channel-4.jpeg">
      </div>

      <div class="video-info">
        <p class="video-title">

          The Simplest Math Problem No One Can Solve- Collatz Conjecture


        </p>
        <p class="video-author">
          Veritasium
        </p>
        <p class="video-stats">
          18M views &#183; 4 months ago
        </p>
      </div>


    </div>
  </div>

  <div class="video-preview">

    <div class="thumbnail-row">
      <img class="thumbnail" src="thumbnails/thumbnail-5.webp">
      <div class="video-time">11:17</div>
    </div>

    <div class="video-info-grid">
      <div class="channel-picture">
        <img class="profile-picture" src="channelpics/channel-5.jpeg">
      </div>

      <div class="video-info">
        <p class="video-title">
          Kadane's Algorithm to Maximum Sum Subarray Problem
        </p>
        <p class="video-author">
          CS Dojo
        </p>
```

```
                <p class="video-stats">
                    519k views &#183; 5 years ago
                </p>
            </div>


        </div>
    </div>

    <div class="video-preview">

        <div class="thumbnail-row">
            <img class="thumbnail" src="thumbnails/thumbnail-6.webp">
            <div class="video-time">19:59</div>
        </div>

        <div class="video-info-grid">
            <div class="channel-picture">
                <img class="profile-picture" src="channelpics/channel-6.jpeg">
            </div>

            <div class="video-info">
                <p class="video-title">
                    Anything You Can Fit In The Circle I'll Pay For
                </p>
                <p class="video-author">
                    MrBeast
                </p>
                <p class="video-stats">
                    141M views &#183; 1 year ago
                </p>
            </div>


        </div>
    </div>


</div>

</body>
</html>
```