PROJECT STAGE 2 – Intro To Database

Today we will know?

What is database?

What are tables?

Basic sql Queries.

Create and Read in CRUD Operation.

What is database?

A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modelling techniques.
Some popular database vendors:MySQL,Oracle Database, PostgreSQL etc.

What are tables?

A table is a collection of related data held in a table format within a database. It consists of columns and rows.
For example in our project there is a table called my_tasks. It contains *four* fields namely *id,task_name,task_date.is_completed*

ID	**TASK_NAME**	**TASK_DATE**	**IS_COMPLETED**
1	My First Task	26 Apr 2020	0

Basic SQL Queries

SQL – Structured Query Language

- *The SQL **SELECT** Statement:* The SELECT statement is used to select data from a database.
- *SQL **INSERT INTO** Statement:* The INSERT INTO statement is used to insert new records in a table.
- *The SQL **UPDATE** Statement:* The UPDATE statement is used to modify the existing records in a table.
- *The SQL **DELETE** Statement:* The DELETE statement is used to delete existing records in a table.

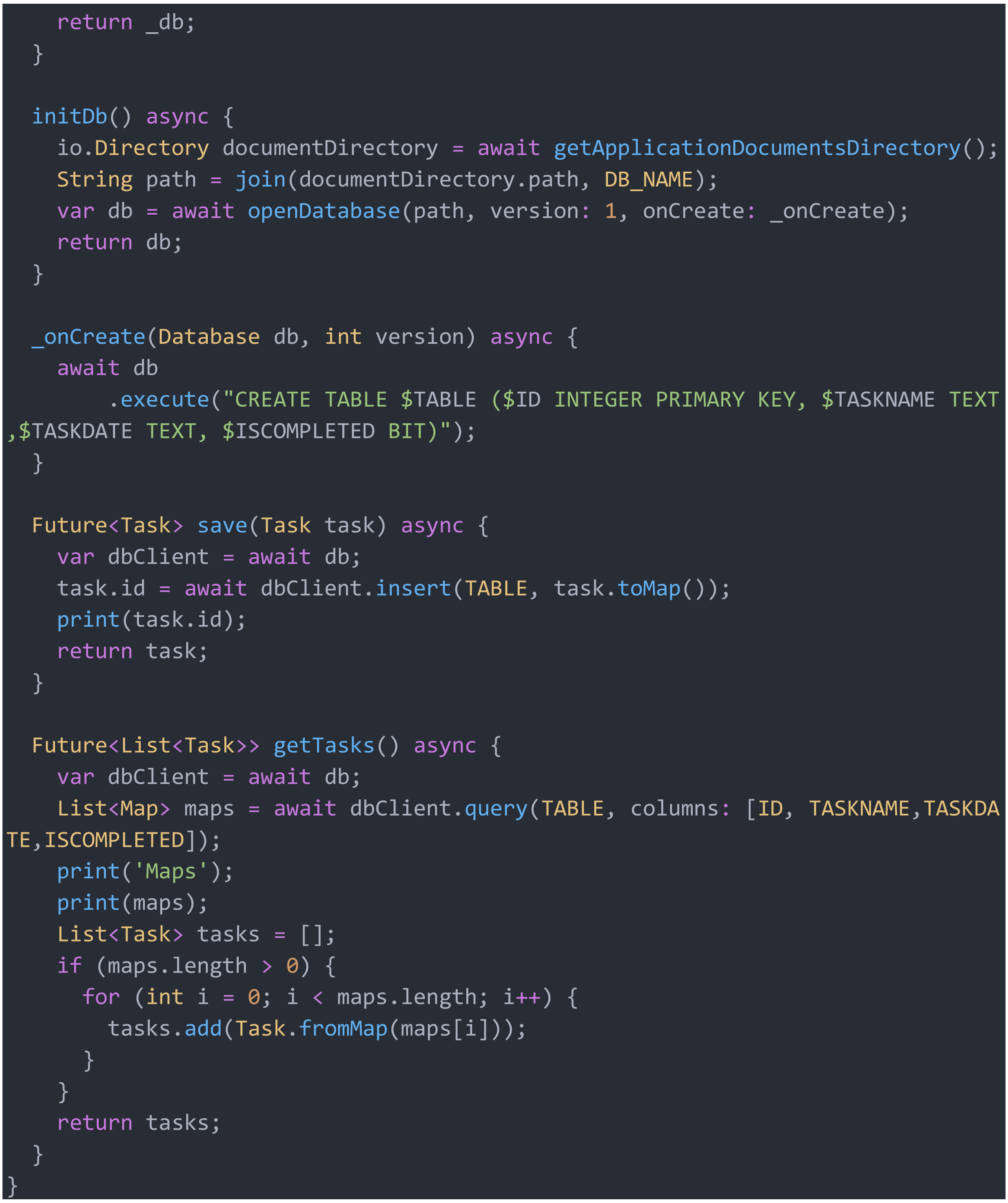
Code: DBHELPER AND TASK

```
class Task{
int id;
  String taskname,taskdate;
  int iscompleted;
  Task(this.id, this.taskname,this.taskdate,this.iscompleted);
  Map<String, dynamic> toMap() {
    var map = <String, dynamic>{
      'id': id,
      'taskname': taskname,
      'taskdate': taskdate,
      'iscompleted': iscompleted,
    };
    return map;
  }

  Task.fromMap(Map<String, dynamic> map) {
    id = map['id'];
    taskname = map['taskname'];
    taskdate = map['taskdate'];
    iscompleted = map['iscompleted'];
  }
}
```

```
import 'dart:async';
import 'dart:io' as io;
import 'package:path/path.dart';
import 'package:sqflite/sqflite.dart';
import 'package:path_provider/path_provider.dart';
import '../model/Task.dart';
class DBHelper {
  static Database _db;
  static const String ID = 'id';
  static const String TASKNAME = 'taskname';
  static const String TASKDATE = 'taskdate';
  static const String ISCOMPLETED = 'iscompleted';
  static const String TABLE = 'Task';
  static const String DB_NAME = 'task.db';

  Future<Database> get db async {
    if (_db != null) {
      return _db;
    }
    _db = await initDb();
```



```
    return _db;
  }

  initDb() async {
    io.Directory documentDirectory = await getApplicationDocumentsDirectory();
    String path = join(documentDirectory.path, DB_NAME);
    var db = await openDatabase(path, version: 1, onCreate: _onCreate);
    return db;
  }

  _onCreate(Database db, int version) async {
    await db
        .execute("CREATE TABLE $TABLE ($ID INTEGER PRIMARY KEY, $TASKNAME TEXT
,$TASKDATE TEXT, $ISCOMPLETED BIT)");
  }

  Future<Task> save(Task task) async {
    var dbClient = await db;
    task.id = await dbClient.insert(TABLE, task.toMap());
    print(task.id);
    return task;
  }

  Future<List<Task>> getTasks() async {
    var dbClient = await db;
    List<Map> maps = await dbClient.query(TABLE, columns: [ID, TASKNAME,TASKDA
TE,ISCOMPLETED]);
    print('Maps');
    print(maps);
    List<Task> tasks = [];
    if (maps.length > 0) {
      for (int i = 0; i < maps.length; i++) {
        tasks.add(Task.fromMap(maps[i]));
      }
    }
    return tasks;
  }
}
```

GEO J VALLAVANCOTTU
geo.j@sjcetpalai.ac.in
8281782564