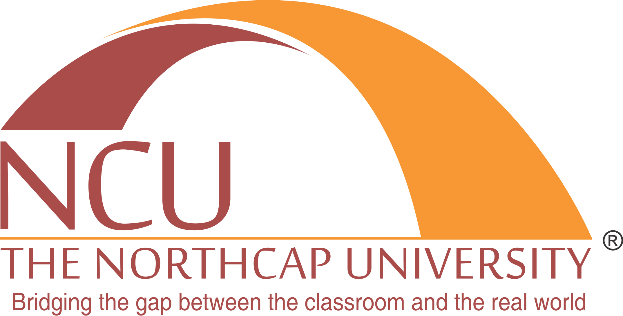


|  |
| --- |
| **Mobile Application Lab** |
| Lab Manual |
| **Department of Computer Science and Engineering The NorthCap University, Gurugram** |

# Mobile Application Lab Manual CSL 276

**Mr. Sumit Kumar**



Department of Computer Science and Engineering NorthCap University, Gurugram- 122001, India Session 2020-21

*Published by:*

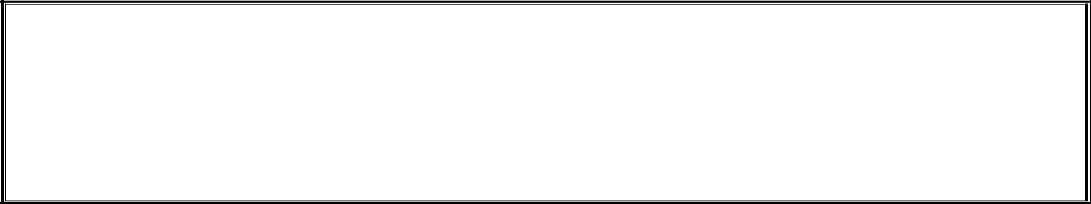
## School of Engineering and Technology Department of Computer Science & Engineering The NorthCap University Gurugram

* **Laboratory Manual is for Internal Circulation only**

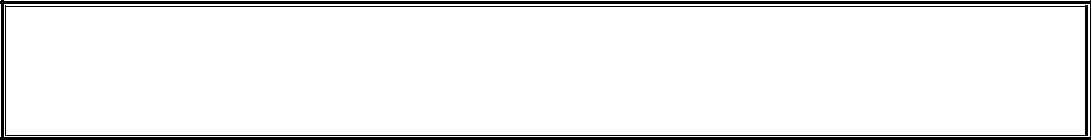
**© Copyright Reserved**

*No part of this Practical Record Book may be*

*reproduced, used, stored without prior permission of The NorthCap University*



Copying or facilitating copying of lab work comes under cheating and is considered as use of unfair means. Students indulging in copying or facilitating copying shall be awarded zero marks for that particular experiment. Frequent cases of copying may lead to disciplinary action. Attendance in lab classes is mandatory.



Labs are open up to 7 PM upon request. Students are encouraged to make full use of labs beyond normal lab hours.

**2020-21**

# PREFACE

Mobile Application (CSL276) Lab Manual is designed to meet the course and program requirements of the NCU curriculum for B.Tech III year students of CSE branch. The concept of the lab work is to give brief practical experience for basic lab skills to students. It provides the space and scope for self-study so that students can come up with new and creative ideas.

The Lab manual is written on the basis of “teach yourself pattern” and expected that students who come with proper preparation should be able to perform the experiments without any difficulty. Brief introduction to each experiment with information about self-study material is provided. The laboratory exercises will include flutter design basics, Dart basics. Further, it includes dynamic Flutter Application Development features and database connectivity. Groups of two shall be created for the students to work on the assigned project. Students are expected to come thoroughly prepared for the lab. General disciplines, safety guidelines and report writing are also discussed.

The lab manual is a part of the curriculum for The NorthCap University, Gurugram. Teacher‟s copy of the experimental results and answer for the questions are available as sample guidelines.

We hope that lab manual would be useful to students of CSE and author requests the readers to kindly forward their suggestions / constructive criticism for further improvement of the work book.

Author expresses deep gratitude to Members, Governing Body-NCU for encouragement and motivation.

## Authors The NorthCap University

**Gurugram, India**

**2020-21**

|  |  |  |
| --- | --- | --- |
| **CONTENTS** | | |
| S.N. | **Details** | Page No. |
|  | **Syllabus** |  |
| 1 | **Introduction** |  |
| 2 | **Lab Requirement** |  |
| 3 | **General Instructions** |  |
| 4 | **List of Experiments** |  |
| 5 | **List of Flip Assignment** |  |
| 6 | **List of Projects** |  |
| 7 | **Rubrics** |  |
| 8 | **Annexure 1 (Format of Lab Report)** |  |
| 9 | **Annexure 2 (Format of Project Report)** |  |

**2020-21**



# SYLLABUS



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Department: | | **Department of CSE & IT** | | | |
| **1. Course Name:**  **SPA Part-I** | | **2. Course Code** | | **3. L-T- P** | **4. Credits** |
| **Code: CSL276** | | 2 - 0 - 4 | 4 |
| **5. Type of Course (Check one):** | | Programme Core | Programme Elective Open Elective | | |
| **6. Frequency of offering (check one):** Odd | | | Even Either Sem. Every Sem. | | |
| **7. Brief Syllabus:**  Introduction to Dart, Dart basics, Flutter Mobile Development Framework, Flutter: Dynamic App development , Database connectivity , Sensor and hardware API call | | | | | |
| **8. Total lecture and Practical Hours for this course**  30 Lectures 60 labs | | | | | |
| **9. Course Outcomes (COs)**  Possible usefulness of this course after its completion i.e. how this course will be practically useful to him once it is completed | | | | | |
| **CO1** | Understanding of Dart Programming Language - Fundamentals and intermediate topics | | | | |
| **CO2** | To understand the function programming and object oriented Programming in DART. | | | | |
| **CO3** | To learn Flutter Mobile Development by building apps, | | | | |
| **CO4** | To design, build, and debug Flutter Android and iOS Apps by fetching data from Server. | | | | |
| **CO5** | Build Flutter apps to communicate with a real-time database and incorporating features which  required Sensor and hardware API calls. | | | | |
| **10. UNIT WISE DETAILS No. of Units:** | | | **5** | | |
| **Unit :I Introduction to flutter & Dart Basics Hours: 6** Introduction to flutter, Understanding flutter Architecture, Dart Basics: Installation, First program in Dart, Data types in Dart: Number, String, Boolean, List, Set, Map; Dart - Const and Final Keywords ; Control Flow Statements: if else, switch case, conditional operators, loop, Labelled loop, Break and  Continue. | | | | | |
| **Unit:II Dart Advance Hours: 6**  Functions in Dart: Functions, Fat Arrow Functions, Optional Positional Parameters, Optional Named Parameters, Functional Programming, var args, Anonymous function, future and Async, Server call, | | | | | |

**2020-21**

|  |
| --- |
| Map & key –value pair, List, Set, Object Oriented Programming in Dart: Class and Object, named constructor, Setters & getters , Is –A , Abstract Class, Interface, import, Down casting; Exception handling. |
| **Unit :III Flutter Design Hours: 6**  Flutter Versions, Flutter architecture, Flutter Future, Flutter MAC and window Setup (Installation of Android Studio), Material Design in Flutter, Building and Understanding First Project , Running the First Project (Android Emulator and IOS Emulator), Widget basics , Types of Widgets, Hello World App with Variable value, Writing Scaffold , AppBar, Adding Row and Column Layout and its Properties, Button Event Binding with function and anonymous function, Stateless v/s Stateful widget. |
| **Unit :IV Flutter: Dynamic App development Hours: 6** Writing EMI App (TextField, Controller, CallBack) Connect Mobile and Test on Mobile, Debugging: Break Points and Logs, Dart Dev Tools. Core Flutter widgets: Scrollable Row and Column, Button and types, ListView, ListTile, Future, Builder , Http Call , Life Cycle GridView, GridTile, AppBar Buttons ,Floating Action Buttons, Modal Bottom Sheet, Themes and Text Styling, Custom Fonts, SizedBox |
| **Unit :V Database connectivity & API Hours: 6**  Routing, Tabs, Drawers, Filters, State management using provider, Connect with FireBase, Login with Gmail, Camera, GPS Access. |
| **11. Guided Project (No. of Hours 10): CRUD App and 7 minute exercise Unguided Project (No. of Hours):NA**  **(Specify NA if not applicable)** |
| **12. Brief Description of Self-learning component by students (through books/resource material etc.): NA** |
| **13. Books Recommended:**  **Text Books**:  **Reference Books**:  **Reference websites: (nptel, swayam, coursera, edx, udemy, lms, official documentation weblink)**  <https://dart.dev/> <https://flutter.dev/> |

## INTRODUCTION

That „learning is a continuous process‟ cannot be over emphasized. It is a more of learning practical concepts with little theoretical knowledge. Thus practical makes an integral part of a learning process.

The purpose of conducting experiments can be stated as follows:

* + - To familiarize the students with the basic concepts of DART, programming skill development and the take home assignments mainly implementation-oriented which have to be coded in DART language. The lab sessions will be based on implementing different concepts on a topics discussed in class.
    - Observing basic structure and characteristics of the Mobile Application Development.
    - Designing of Mobile applications.
    - Hands on experience on the mobile applications development using DART.

## LAB REQUIREMENTS

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Requirements** | **Details** |
| **1** | **Software Requirements** | Android studio, DART package, Flutter SDK |
| **2** | **Operating System** | Windows 64 bit |
| **3** | **Hardware Requirements** | CPU-I5 with 8GB RAM, 5 GB HDD space, Mac machine/ phone to test the application |
| **4** | **Required Bandwidth** | 5Mbps for updates and online help |

* 1. **GENERAL INSTRUCTIONS**
  2. **General discipline in the lab**
     + Students must turn up in time and contact concerned faculty for the experiment they are supposed to perform.
     + Students will not be allowed to enter late in the lab.
     + Students will not leave the class till the period is over.
     + Students should come prepared for their experiment.
     + Experimental results should be entered in the lab report format and certified/signed by concerned faculty/ lab Instructor.
     + Students must get the connection of the hardware setup verified before switching on the power supply.
     + Students should maintain silence while performing the experiments. If any necessity arises for discussion amongst them, they should discuss with a very low pitch without disturbing the adjacent groups.
     + Violating the above code of conduct may attract disciplinary action.
     + Damaging lab equipment or removing any component from the lab may invite penalties and strict disciplinary action.

## Attendance

* + - Attendance in the lab class is compulsory.
    - Students should not attend a different lab group/section other than the one assigned at the beginning of the session.
    - On account of illness or some family problems, if a student misses his/her lab classes, he/she may be assigned a different group to make up the losses in consultation with the concerned faculty / lab instructor. Or he/she may work in the lab during spare/extra hours to complete the experiment. No attendance will be granted for such case**.**

## Preparation and Performance

* + - Students should come to the lab thoroughly prepared on the experiments they are assigned to perform on that day. Brief introduction to each experiment with information about self-study reference is provided on LMS.
    - Students must bring the lab report during each practical class with written records of the last experiments performed complete in all respect.
* Each student is required to write a complete report of the experiment he has performed and bring to lab class for evaluation in the next working lab. Sufficient space in work book is provided for independent writing of theory, observation, calculation and conclusion.
* Students should follow the Zero tolerance policy for copying / plagiarism. Zero marks will be awarded if found copied. If caught further, it will lead to disciplinary action.
* Refer **Annexure 1** for Lab Report Format.

## LIST OF EXPERIMENTS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Title of the Experiment** | **Software used** | **Unit**  **covered** | **CO**  **Covered** | **Time**  **Required** |
| 1. | Visit [pub.dev](http://pub.dev/) and try using some packages like ImagePicker to take photos from Camera & Gallery. | Android studio, DART package,  Flutter SDK | 1 | CO1 | 4hrs |
| 2. | Develop a Simple EMI Calculator using Pie chart and TextFields. | Android studio, DART package, Flutter SDK | 1 | CO1 | 4hrs |
| 3. | Develop an application that allows users to login using Email & Password. (Use firebase for OAuth functionality) | Android studio, DART package, Flutter SDK | 2 | CO2 | 4hrs |
| 4. | To Develop an Audio player that has the following functionality: play, pause, skip ahead/behind. | Android studio, DART package, Flutter SDK | 3 | CO3 | 4hrs |
| 5. | To Develop a simple UI using ListView (Like Book My Show). | Android studio, DART package,  Flutter SDK | 3 | CO3 | 4hrs |
| 6. | Build a simple application that fetches data from Cloud Firestore like Products from FireBaseFireStore. | Android studio, DART package, Flutter SDK | 4 | CO4 | 4hrs |
| 7. | To Develop a simple Register Page (Form) using rows and column widgets. | Android studio,  DART package, Flutter SDK | 4 | CO4 | 4hrs |

**LIST OF FLIP EXPERIMENT**

1. Design a To Do List Application
2. Design a Time sheet Application.
3. Design an E-commerce Application
4. Design an Application for Blood Bank Management system
5. Design a School Fee Management System Application
6. Design a School Bus Transport System Application

## PROJECT

**To design an interface which is used to conduct the test for students across the globe while sitting at their home.**

* **This engine is divided into three modules :-**
  1. **Admin Module (For administrator )**
  2. **Teacher Module ( For Teaching staff )**
  3. **Student Module ( For student’s)**
* **The test engine will help for contact free examination and in mock test too. Keeping in mind of the current situation it will help in the regular assessment of the student’s.**
* **Technology Stack:**
  + **Flutter: To design the front end of an application**
  + **MySQL: For database connectivity.**
* **Admin’s module overview**

Admin module is for the administrator, who review the activities of all the other users.

Basically, it is for creating the user‟s role and right and giving them particular access according to their roles.

Admin can view the number of tests created, number of students and highest scores, the number of groups and student progress graph and many more.

## Features list

* + Can login using default userid and password saved in DB.
  + Can view the number of tests created.
  + Can view number of students.
  + Can create the role and rights of user.
  + Can create subjects.
  + Can view number of groups
  + Can view user‟s progress graph

## Teacher’s module overview

* + **Teacher module is for the teaching staff or the examination staff , who can create the test for the student’s**
  + **Basically, it is for creating the test and deploying it to the respective student’s**
  + **More that one student or a group of student can also me given test.**
* **Features list**
  + Teacher can register themselves.
  + Login of the teacher using data from DB.
  + Creating of test.
  + Can create question.
  + Group can be created.
  + Assign test to respective members.
  + View added questions
  + MCQ type of questions can created .

## Student’s module overview

* Student module is for the students , who can attempt the assigned tests allotted to them by their teachers.
* While attempting a test they can go to next or previous question according to their choice and after attempting can submit it
* They can view their results and score cards after giving a test.

## Features list

* + **Students can register themselves.**
  + **Login of the student using data from DB.**
  + **Attempting assigned test.**
  + **Can add themselves to specific group using registration code.**
  + **Can go to previous or next question according to their choices.**
  + **After submitting test can view their scores.**

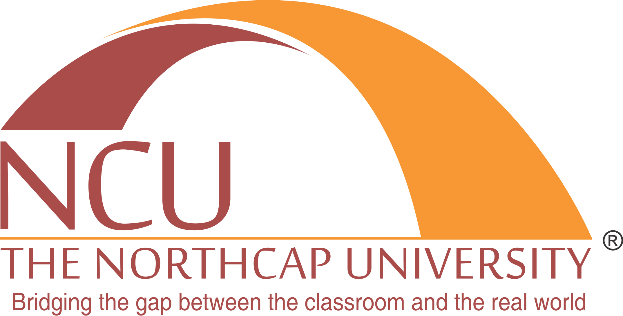
1. **RUBRICS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.**  **No.** | **TYPE**  **OF COURSE** | **PARTICULAR** | **ALLOTTED**  **RANGE OF MARKS** | **PASS CRITERIA** |
| 1 | Project Based Course (L-T-P/L-  T-0/L-0-  P/L-0-0) | End Term Project | 40% | Must Secure 30% Marks Out of |
| Major Test | 35% |
| Class Test/ Assignment | 15% |
| Class Participation Evaluation Through Class | 10% |

**Annexure 1**

**Mobile Application Lab Manual (CSL 276)**

**Lab Practical Report**



Faculty name: Student name:

Roll No.: Semester: Group:

## Department of Computer Science and Engineering NorthCap University, Gurugram- 122001, India Session 2020-21

**Manish Kumar Sharma**

**Roll Number:- 19csu173**

**2020-21**

## INDEX

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S. No.** | **Experiment** | **Page No.** | **Date of Experiment** | **Date of Submission** | **Marks** | **CO**  **Covered** | **Sign** |
| **1** | Visit [pub.dev](http://pub.dev/) and try using some packages like ImagePicker to take photos from Camera & Gallery. |  |  |  |  |  |  |
| **2** | Develop a Simple EMI Calculator using Pie chart and TextFields. |  |  |  |  |  |  |
| **3** | Develop an application that allows users to login using Email & Password. (Use firebase for OAuth functionality) |  |  |  |  |  |  |
| **4** | To Develop an Audio player that has the following functionality: play, pause, skip ahead/behind. |  |  |  |  |  |  |
| **5** | To Develop a simple UI using ListView (Like Book My Show). |  |  |  |  |  |  |
| **6** | Build a simple application that fetches data from Cloud Firestore like Products from FireBaseFireStore. |  |  |  |  |  |  |
| **7** | To Develop a simple Register Page (Form) using rows and column widgets. |  |  |  |  |  |  |

## EXPERIMENT NO. 1

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Date:** |
| **GitHub Link:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective(s):**  Visit [pub.dev](http://pub.dev/) and try using some packages like ImagePicker to take photos from Camera & Gallery. |
| **Outcome:**  The students will be able to include different widgets in the application and can make them enabled camera and Gallery which are the most basic features of any application now a days. |
| **Background Study:**  Pub is the package manager for the Dart programming language, containing reusable libraries & packages for Flutter, AngularDart, and general Dart programs.  **Camera Plugin:**  A Flutter plugin for iOS and Android allowing access to the device cameras.  **Features:**   * Display live camera preview in a widget. * Snapshots can be captured and saved to a file. * Record video. |

|  |
| --- |
| * Add access to the image stream from Dart.   **Installation**  First, add camera as a [dependency in your pubspec.yaml file](https://flutter.dev/using-packages/).  **Features:**   * Display live camera preview in a widget. * Snapshots can be captured and saved to a file. * Record video. * Add access to the image stream from Dart.   **Installation**  First, add camera as a [dependency in your pubspec.yaml file](https://flutter.dev/using-packages/). [Location](https://pub.dev/packages/location)  A Flutter plugin to easily handle realtime location in iOS and Android. Provides settings for optimizing  performance or battery.  A Flutter geolocation plugin which provides easy access to platform specific location services ([FusedLocationProviderClient](https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient) or if not available the [LocationManager](https://developer.android.com/reference/android/location/LocationManager) on Android  and [CLLocationManager](https://developer.apple.com/documentation/corelocation/cllocationmanager) on iOS).  **Features**   * Get the last known location; * Get the current location of the device; * Get continuous location updates; * Check if location services are enabled on the device; * Calculate the distance (in meters) between two geocoordinates; * Calculate the bearing between two geocoordinates;   **Usage**  To add the geolocator to your Flutter application read the [install](https://pub.dev/packages/geolocator/install) instructions. Below are some Android and iOS specifics that are required for the geolocator to work correctly. |

AndroidiOSWeb

## API

**Geolocation**

To query the current location of the device simply make a call to the getCurrentPosition method. You can finetune the results by specifying the following parameters:

* + desiredAccuracy: the accuracy of the location data that your app wants to receive;
* timeLimit: the maximum amount of time allowed to acquire the current location. When the time limit is passed a TimeOutException will be thrown and the call will be cancelled. By default no limit is configured.

**import** 'package:geolocator/geolocator.dart';

Position position = **await** Geolocator.getCurrentPosition(desiredAccuracy: LocationAccuracy.high);

To query the last known location retrieved stored on the device you can use

the getLastKnownPosition method (note that this can result in a null value when no location details are available):

**import** 'package:geolocator/geolocator.dart';

Position position = **await** Geolocator.getLastKnownPosition();

To listen for location changes you can call the getPositionStream to receive stream you can listen to and

receive position updates. You can finetune the results by specifying the following parameters:

* + desiredAccuracy: the accuracy of the location data that your app wants to receive;
  + distanceFilter: the minimum distance (measured in meters) a device must move horizontally before an update event is generated;
  + timeInterval: (Android only) the minimum amount of time that needs to pass before an update event is generated;
  + timeLimit: the maximum amount of time allowed between location updates. When the time limit is passed a TimeOutException will be thrown and the stream will be cancelled. By default no limit is configured.

## Settings

In some cases it is necessary to ask the user and update their device settings. For example when the user initially permanently denied permissions to access the device's location or if the location services are not enabled (and, on Android, automatic resolution didn't work). In these cases you can use the openAppSettings or openLocationSettings methods to immediately redirect the user to the device's settings page.

**2020-21**

|  |
| --- |
| On Android the openAppSettings method will redirect the user to the App specific settings where the user can update necessary permissions. The openLocationSettings method will redirect the user to the location settings where the user can enable/ disable the location services.  On iOS we are not allowed to open specific setting pages so both methods will redirect the user to the Settings App from where the user can navigate to the correct settings category to update permissions or enable/ disable the location services.  **import** 'package:geolocator/geolocator.dart';  **await** Geolocator.openAppSettings();  **await** Geolocator.openLocationSettings();  **import** 'package:geolocator/geolocator.dart';  **sound\_stream**  *This plugin is still in early development stage*  A Flutter plugin started from my own needs: Stream audio data from Mic and data to Audio engine without using a file. We can use it to stream audio via network or use it with STT/TTS functions.  Current features:   * Provides stream of data from mic (Uint8List) * Player that receive stream of raw sound data (Uint8List) * Support both Android and iOS (cross-platform) * Recorder & Player can work simultaneously Limitations: * Only support PCM 16bit Mono (for now) * Data type send/received from stream must be Uint8List. ([Because of this Flutter's limitation](https://flutter.dev/docs/development/platform-integration/platform-channels?tab=ios-channel-swift-tab&codec)) To-do list: * Support more audio formats * Support more platforms (Windows, macOS, Web) * Current code might be messy. Should clean it up (someday) |
| **Program Code:** |

## Question Bank:

1. What is Dart?
   1. Android Development Kit
   2. Web Development Kit
   3. Programming Language
   4. SDK to build beautiful IOS, Android, Web & Desktop Native Apps
2. What is Flutter?
   1. Android Development Kit
   2. Web Development Kit
   3. Programming Language
   4. SDK to build beautiful IOS, Android, Web & Desktop Native Apps
3. Which programming language is used by Flutter?
   1. Ruby
   2. Dart
   3. C++
   4. Kotlin
4. Who created flutter?
   1. Facebook
   2. Adobe
   3. Google
   4. Microsoft
5. Row displays its children in a array.
   1. Horizontal
   2. Vertical
6. Column displays its children in a array.
   1. Horizontal
   2. Vertical
7. By default, Row doesn‟t support scrolling.
   1. True
   2. False
8. By default, Column doesn‟t support scrolling.
   1. True
   2. False
9. If there are numerous child widgets, it is preferred to use ListView.
   1. True
   2. False

## Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

## EXPERIMENT NO. 2

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Date:** |
| **GitHub Link:** |
| **Faculty Signature:** |
| **Marks:** |
| **Student Name and Roll Number: Paras Bhatia, 18csu153** |

|  |
| --- |
| **Objective:**  Develop a Simple EMI Calculator using Pie chart and TextFields. |
| **Outcome:**  To design a User Friendly application with multiple widgets. |
| **Background Study:**  **SliverAppBar class**  A material design app bar that integrates with a [CustomScrollView](https://api.flutter.dev/flutter/widgets/CustomScrollView-class.html).  An app bar consists of a toolbar and potentially other widgets, such as a [TabBar](https://api.flutter.dev/flutter/material/TabBar-class.html) and  a [FlexibleSpaceBar.](https://api.flutter.dev/flutter/material/FlexibleSpaceBar-class.html) App bars typically expose one or more common actions with [IconButton](https://api.flutter.dev/flutter/material/IconButton-class.html)s which are optionally followed by a [PopupMenuButton](https://api.flutter.dev/flutter/material/PopupMenuButton-class.html) for less common operations.  liver app bars are typically used as the first child of a [CustomScrollView](https://api.flutter.dev/flutter/widgets/CustomScrollView-class.html), which lets the app bar integrate with the scroll view so that it can vary in height according to the scroll offset or float above the other content in the scroll view. For a fixed-height app bar at the top of the screen  see [AppBar,](https://api.flutter.dev/flutter/material/AppBar-class.html) which is used in the [Scaffold.appBar](https://api.flutter.dev/flutter/material/Scaffold/appBar.html) slot.  The AppBar displays the toolbar widgets, [leading,](https://api.flutter.dev/flutter/material/SliverAppBar/leading.html) [title](https://api.flutter.dev/flutter/material/SliverAppBar/title.html), and [actions](https://api.flutter.dev/flutter/material/SliverAppBar/actions.html), above the [bottom](https://api.flutter.dev/flutter/material/SliverAppBar/bottom.html) (if any). If a [flexibleSpace](https://api.flutter.dev/flutter/material/SliverAppBar/flexibleSpace.html) widget is specified then it is stacked behind the toolbar and the bottom widget. |
| **Program Code:** |

|  |
| --- |
| **Question Bank:**   1. Which of these are scrollable widgets?    1. ListView    2. GridView    3. Both of the above    4. None of the above 2. Which is a scrollable and 2D array of widgets?    1. ListView    2. GridView    3. Row    4. Container 3. Which of these constructors of the GridView creates a scrollable, 2D array of widgets with a fixed number of tiles in the cross axis?    1. GridView()    2. GridView.extent()    3. GridView.count()    4. GridView.builder() 4. Which of these constructors of the GridView creates a scrollable, 2D array of widgets with tiles that each has a maximum cross-axis event?    1. GridView()    2. GridView.extent()    3. GridView.count()    4. GridView.builder() 5. Which of the following property of GridView controls how one widget replaces another widget in the tree?    1. key    2. padding    3. primary    4. anchor 6. Which of the following methods of GridView describes the part of the user interface represented by this widget?    1. buildChildLayout    2. buildSlivers    3. build    4. buildViewPort |

**Student Work Area**

Algorithm/Flowchart/Code/Sample Outputs

## EXPERIMENT NO. 3

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Date:** |
| **GitHub Link:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Develop an application that allows users to login using Email & Password. (Use firebase for OAuth functionality) |
| **Outcome:**  The students will able to create an application where the features of login are enabled using Firebase. |
| **Background Study:**  Once installed, you can access the [firebase\_auth](https://pub.dev/documentation/firebase_auth/latest/firebase_auth/firebase_auth-library.html) plugin by importing it in your Dart code:  *import* 'package:firebase\_auth/firebase\_auth.dart';  Before using Firebase Auth, you must first have ensured you have [initialized FlutterFire](https://firebase.flutter.dev/docs/overview#initializing-flutterfire). To create a new Firebase Auth instance, call the [instance](https://pub.dev/documentation/firebase_auth/latest/firebase_auth/FirebaseAuth/instance.html) getter on [FirebaseAuth](https://pub.dev/documentation/firebase_auth/latest/firebase_auth/FirebaseAuth-class.html):  FirebaseAuth auth = FirebaseAuth.instance;  By default, this allows you to interact with Firebase Auth using the default Firebase App used whilst installing FlutterFire on your platform. If however you'd like to use a secondary Firebase App, use the [instanceFor](https://pub.dev/documentation/firebase_auth/latest/firebase_auth/FirebaseAuth/FirebaseAuth.instanceFor.html) method:  FirebaseApp secondaryApp = Firebase.app('SecondaryApp'); FirebaseAuth auth = FirebaseAuth.instanceFor(app: secondaryApp); |

**2020-21**

|  |
| --- |
| **Program Code:** |
| **Question Bank:**   1. What is a portion of a scrollable area that you can define to behave in a special behaviour?    1. Sliver    2. SliverAppBar    3. SliverGrid    4. SliverList 2. What places multiple box children in a linear array along the main axis?    1. Sliver    2. SliverAppBar    3. SliverGrid    4. SliverList 3. What places multiple box children in a two dimensional arrangement?    1. Sliver    2. SliverAppBar    3. SliverGrid    4. SliverList 4. What is a material design app bar that integrates with a CustomScrollView?    1. Sliver    2. SliverAppBar    3. SliverGrid    4. SliverList 5. In SliverList class, Each child is forced to have the in the cross axis.    1. Sliver.crossAxisExtent    2. SliverConstraints.crossAxisExtent    3. SliverConstraints.crossExtent    4. SliverConstraints.AxisExtent 6. determines its scroll offset by “dead reckoning”.    1. Sliver    2. SliverAppBar    3. SliverGrid    4. SliverList 7. places its children in arbitrary positions determined by gridDelegate.    1. Sliver    2. SliverAppBar    3. SliverGrid    4. SliverList |

|  |
| --- |
| 1. are typically used as the first child of a CustomScrollView.    1. Sliver    2. SliverAppBar    3. SliverGrid    4. SliverList 2. Which widget that clips its child using a path?    1. Sliver    2. SliverAppBar    3. ClipPath    4. SliverList |

## Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

## EXPERIMENT NO. 4

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Date:** |
| **GitHub Link:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  To Develop an Audio player that has the following functionality: play, pause, skip ahead/behind. |
| **Outcome:**  The students will able to create their own media player by learning its different functionality. |
| **Background Study:**  This recipe demonstrates how to use the video\_player package to stream a video from the internet with basic play and pause controls using the following steps:   1. Add the video\_player dependency. 2. Add permissions to your app. 3. Create and initialize a VideoPlayerController. 4. Display the video player. 5. Play and pause the video.   **1. Add the video\_player dependency**  This recipe depends on one Flutter plugin: video\_player. First, add this dependency to your pubspec.yaml.  *content\_copy*  dependencies: flutter:  sdk: flutter video\_player: |

|  |
| --- |
| 1. **Add permissions to your app**   Next, update your android and ios configurations to ensure that your app has the correct permissions to stream videos from the internet.  **iOS**  For iOS, add the following to the Info.plist file found at <project root>/ios/Runner/Info.plist.  *content\_copy*  <key>NSAppTransportSecurity</key>  <dict>  <key>NSAllowsArbitraryLoads</key>  <true/>  </dict>   1. **Create and initialize a VideoPlayerController**   Now that you have the video\_player plugin installed with the correct permissions, create  a VideoPlayerController. The VideoPlayerController class allows you to connect to different types of videos and control playback.  Before you can play videos, you must also initialize the controller. This establishes the connection to the video and prepare the controller for playback.  To create and initialize the VideoPlayerController do the following:   * 1. Create a StatefulWidget with a companion State class   2. Add a variable to the State class to store the VideoPlayerController   3. Add a variable to the State class to store the Future returned from VideoPlayerController.initialize   4. Create and initialize the controller in the initState method   5. Dispose of the controller in the dispose method  1. **Display the video player**   Now, display the video. The video\_player plugin provides the [VideoPlayer](https://pub.dev/documentation/video_player/latest/video_player/VideoPlayer-class.html) widget to display the video initialized by the VideoPlayerController. By default, the VideoPlayer widget takes up as much space as possible. This often isn‟t ideal for videos because they are meant to be displayed in a specific aspect ratio, such as 16x9 or 4x3.  Therefore, wrap the VideoPlayer widget in an [AspectRatio](https://api.flutter.dev/flutter/widgets/AspectRatio-class.html) widget to ensure that the video has the correct proportions.  Furthermore, you must display the VideoPlayer widget after  the \_initializeVideoPlayerFuture() completes. Use FutureBuilder to display a loading spinner |

**2020-21**

|  |
| --- |
| until the controller finishes initializing. Note: initializing the controller does not begin playback.  **5. Play and pause the video**  By default, the video starts in a paused state. To begin playback, call the [play()](https://pub.dev/documentation/video_player/latest/video_player/VideoPlayerController/play.html) method provided by the VideoPlayerController. To pause playback, call the [pause()](https://pub.dev/documentation/video_player/latest/video_player/VideoPlayerController/pause.html) method.  For this example, add a FloatingActionButton to your app that displays a play or pause icon depending on the situation. When the user taps the button, play the video if it‟s currently paused, or pause the video if it‟s playing. |
| **Program Code:** |
| **Question Bank:**   1. Which plugin is used to access and control camera?    1. microphone    2. camera    3. location    4. internet 2. Which plugin is used to access and record audios?    1. microphone    2. camera    3. location    4. internet 3. Which plugin is used to access location?    1. microphone    2. camera    3. location    4. internet 4. Which plugin is used for http requests?    1. http proxy    2. japx    3. http    4. alice 5. Which plugin is used to integrate google analytics for firebase?    1. firebase\_ml\_vision    2. firebase\_storage    3. firebase\_admob    4. firebase\_analytics |

## Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

## EXPERIMENT NO. 5

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Date:** |
| **GitHub Link:** |
| **Faculty Signature:** |
| **Marks:** |
| **Student Name and Roll Number: Paras Bhatia, 18csu153** |

|  |
| --- |
| **Objective:**  To Develop a simple UI using ListView (Like Book My Show). |
| **Outcome:**  The students will be able to understand that how to include ListView and GridView in applications. |

|  |
| --- |
| **Background Study:**  Image for post  **List**  The list is basically an arrangement of items in such a way that they look organized and easily accessible. The flutter list is more than just a normal list. Here are some types of lists available in flutter:   1. Grid List 2. Horizontal List   Mobile application‟s fundamental display pattern is showing data in the form of a list. We can use a standard **ListView** constructor to achieve this goal for a fewer item containing list. **ListView** constructor has **ListTile** widget  **ListTile:** A list tile contains one to three lines of text optionally flanked by icons or other widgets, such as checkboxes. The icons (or other widgets) for the tile are defined with the leading and trailing parameters.  Here in LIstTile two very important widgets are **leading** and **trailing**  **leading:** This widget is used to show something like an icon or logo or image in the left most of the list.  **trailing:** This widget is used to show items like icon, image or logo right most of the list. ListView(  children: <Widget>[ ListTile(  leading: FlutterLogo(),  trailing: Icon(Icons.*more\_vert*),  title: Text('One-line with leading & trailing widget'),  )  ],  )  **Grid List**  An alternative to the normal List is Grid List. If you want to show items in a list as a grid that comes one after another below as well as side by side, GridView is the widget for you. Let's see how it works  **GridView:** GridView is a scrollable 2D array of widgets. The most frequently used grid layout is ***GridView.count***  GridView.count |

**2020-21**

|  |
| --- |
| *It creates a 2D scrollable widget with the flexible number of tiles in the cross axis.*  GridView.count(  ),  crossAxisCount  It gives you the flexibility to show the number of grids in the cross axis of your view. The below-given snippet will allow showing 2 grid side by side on the body area. crossAxisCount: 2  List.generate  It gives you the power of replication of the same grid to a number of times you want to generate it.  **Example** GridView.count( crossAxisCount: 2 ,  children: List.generate(50,(index){ return Container(  child: Card(  color: Colors.*blue*,  ),  );  }),  ) |
| **Program Code:** |
| **Question Bank:**   1. In which file the video dependency will be added?    1. AndroidManifest.xml    2. Stateful Class    3. Stateless Class    4. pubspec.yaml 2. What is the location for AndroidManifest.xml file?    1. android/app/src/main/AndroidManifest.xml    2. app/src/main/AndroidManifest.xml    3. <project root>/android/app/src/main/AndroidManifest.xml    4. src/main/AndroidManifest.xml 3. In which file will the permissions be added?    1. AndroidManifest.xml    2. Stateful Class    3. Stateless Class |

|  |
| --- |
| d) pubspec.yaml   1. By default, Video Editor takes how much space? a) 16\*9   b) 4\*3   * 1. No default space (Always need to define)   2. As much space as possible  1. Which plugin is used to play videos? 2. video 3. video\_player 4. play\_video 5. player 6. On IOS, the video\_player plugin uses \_    1. AVPlayer    2. VideoPlayer    3. AudioPlayer    4. Player 7. By default, Video starts in a state.    1. played    2. paused 8. For IOS to grant permissions to play videos, add the following to the    1. AndroidManifest.xml    2. pubspec.yaml    3. Info.plist    4. Info.list |

## Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

## EXPERIMENT NO. 6

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Date:** |
| **GitHub Link:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Build a simple application that fetches data from Cloud Firestore like Products from FireBaseFireStore. |
| **Outcome:**  The students will be able to understand Build a simple application that fetches data from Cloud Firestore. |
| **Background Study:**  Paginate data with query cursors  With query cursors in Cloud Firestore, you can split data returned by a query into batches according to the parameters you define in your query.  Query cursors define the start and end points for a query, allowing you to: Return a subset of the data.  Paginate query results.  However, to define a specific range for a query, you should use the where() method describe. |
| **Program Code:** |

|  |
| --- |
| **Question Bank:**   1. The package provides the simplest way to fetch data from the internet.    1. internet    2. package    3. camera    4. http 2. Additionally, in your AndroidManifest.xml file add the permission.    1. internet    2. package    3. camera    4. http 3. http.get() method returns a that contains Response.    1. int    2. Future    3. boolean    4. character 4. Convert the response into a .    1. int    2. string    3. custom dart object    4. character 5. A function that tells Flutter what to render, depending on the state of the Future.    1. fetch    2. builder    3. get    4. response 6. snapshot.hasdata only returns when the snapshot contains a non-null data value.    1. true    2. false 7. To fetch the data, Call the method in initstate() method.    1. fetch()    2. builder()    3. get()    4. response() |

**Student Work Area**

Algorithm/Flowchart/Code/Sample Outputs

## EXPERIMENT NO. 7

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Date:** |
| **GitHub Link:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  To Develop a simple Register Page (Form) using rows and column widgets. |
| **Outcome:**   * The students will understand the need of functions, how to declare and define functions in C. * Students will learn the two methods of function calling – call by value and call by reference. * Students will be acquainted with the concept of recursion, their declaration and definition.. |
| **Background Study:**   * Row and Column are the two most important and powerful widgets in Flutter. These widgets let you align children horizontally and vertically as per the requirement. As we know that when we design any UI(User Interface) in flutter, we need to arrange its content in the Row and Column manner so these Row and Column widgets are required when designing UI. * Properties of Row and Column Widgets: * **children:** This property takes in List<Widget>, that is a list of widgets to display inside the Row or the Column widget. * **clipBehaviour:** This property holds Clip class as the object to decide whether the content on the Row or Column should be clipped or not. * **crossAxisAlignment:** The crossAxisAlignment takes in CrossAxisAlignment enum as the   object to how the children‟s widgets should be places in crossAxisAlignment. For Row it is |

**2020-21**

|  |
| --- |
| vertical and for Column it is horizontal.   * **direction:** This property holds as the Axis enum object to decide the direction used in the main axis. For Row and Column, it is fixed. * **mainAxisAlignment:** This property takes in MainAxisAlignment enum as the object to decide how the children widgets should be place in mainAxisAlignment. For Row it is horizontal and for Column it is vertical. * **mainAxisSize:** This property decides the size of main-axis by taking in MainAxisSize enum as the object. * **runtimeType:** This property tells the run-time type of the Row or Column widget. * **textBaseline:** This property is responsible for the alignment of the text in the Row or Column widget with respect to a baseline. * **textDirection:** This property controls the text direction of the Row or Column widget, which can either be from left-to-right (by default) or right-to-left. * **verticalDirection:** This property takes in Vertical Direction enum as the object to determine the order in which the children should be layered. |
| **Program Code:** |
| **Question Bank:**   1. Which library is used for authentication with a remote service on behalf of a user, making authorized HTTP requests with the user‟s credentials?    1. internet    2. package    3. oauth    4. http 2. Launch a WebView inside the app and listen for a redirect using \_.    1. webview\_flutter    2. url\_launcher    3. uni\_links    4. redirect 3. Launch a browser using url\_launcher and listen for a redirect using .    1. webview\_flutter    2. url\_launcher    3. uni\_links    4. redirect |

|  |
| --- |
| 1. A Flutter plugin for launching a URL in the mobile platform.    1. webview\_flutter    2. url\_launcher    3. uni\_links    4. redirect 2. The latest OAuth 2.0 Security Best Current Practice disallows the password grant   .   * 1. partially   2. easily   3. securily   4. entirely |

## Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

## Annexure 2

**Programming for Data Science CSL225**

Project Report



Faculty name: Student name:

Roll No.: Semester: Group:

Department of Computer Science and Engineering The NorthCap University, Gurugram- 122001, India Session 2020-21

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S.No** |  | **Page No.** |
| **1.** | **Project Description** |  |
| **2.** | **Problem Statement** |  |
| **3.** | **Analysis**   * 1. **Hardware Requirements**   2. **Software Requirements** |  |
| **4.** | **Design**   * 1. **Data/Input Output Description:**   2. **Algorithmic Approach / Algorithm / DFD / ER diagram/Program Steps** |  |
| **5.** | **Implementation and Testing (stage/module wise)** |  |
| **6.** | **Output (Screenshots)** |  |
| **7.** | **Conclusion and Future Scope** |  |