





STIW2044: Mobile Programming  
Semester A212  
School of Computing, CAS, UUM

## FRONT COVER

### MIDTERM EXAM

<b>Name</b>	Juanrico Alvaro
<b>Matric No</b>	702301
<b>YouTube Presentation Link</b>	<a href="https://youtu.be/6mW5gURV-7I">https://youtu.be/6mW5gURV-7I</a>
<b>Phone Number</b>	0136798361
<b>GitHub Link</b>	<a href="https://github.com/victorico123/flutter-projct-UUM">https://github.com/victorico123/flutter-projct-UUM</a>
<b>Submission Date</b>	11/6/2022
<b>Acknowledgment</b>	I hereby acknowledge that the following works are from my effort in submitting this document. If found otherwise, severe action such as marks deduction or removal from the assignment can be taken against me.
<b>Digital Signature</b>	
<b>Students Picture</b>	
<b>Email use during book purchase</b>	juanricoalvaro@gmail.com
<b>Your digital key from the book purchase</b>	5oRiMDDWKW4U5hz

**A212 MOBILE PROGRAMMING (100 MARKS)****Instructions**

The following midterm assignment is a continuation of lab assignment 2.

You have been assigned to continue with the development "My Tutor" mobile application from lab assignment 2. Complete the following instructions by implementing the solution using the Flutter-PHP-MySQL development stacks. You may use the lab 2 assignment project to complete this midterm assignment.

1. User login module: The following module will require for implementing user login.
  - a. Once the user registration module is completed and the user successfully registers their account, navigate to the login screen and perform the user authentication. Implement backend service for user authentication. Fill in the following section with the required codes and scripts.
  - i. User class definition

5 Marks

```
class User {
  int? id;
  String? name;
  String? email;
  String? phone;
  String? address;
  String? image;

  User({this.id, this.email, this.name, this.phone, this.address,
this.image});

  User.fromJson(Map<String, dynamic> json) {
    id = json['id'];
    email = json['email'];
    name = json['name'];
    phone = json['phone'];
    address = json['address'];
    image = json['image'];
  }

  Map<String, dynamic> toJson() {
    final Map<String, dynamic> data = <String, dynamic>{};
    data['id'] = id;
    data['email'] = email;
    data['name'] = name;
```

```
data['phone'] = phone;
data['address'] = address;
data['image'] = image;
return data;
}
}
```

ii. PHP backend script

10 Marks

```
<?php
if (!isset($_POST)) {
    $response = array('status' => 'failed', 'data' => null);
    sendJsonResponse($response);
    die();
}
include_once("dbconnect.php");
$email = addslashes($_POST['email']);
$password = $_POST['password'];
$sqlLogin = "SELECT * FROM users WHERE email = ? AND password = ?";
$stmt = $conn->prepare($sqlLogin);
$stmt->bind_param("ss", $email, $password);

if ($stmt->execute()) {
    $result = $stmt->get_result();
    $user_name = "";
    while ($row = $result->fetch_assoc()) {
        $user['id'] = $row['id'];
        $user['email'] = $row['email'];
        $user['name'] = $row['name'];
        $user['phone'] = $row['phone'];
        $user['address'] = $row['address'];
        $user['image'] = $row['image'];
    }
    $response = array('status' => 'success', 'data' => $user);
    sendJsonResponse($response);
} else {
    $response = array('status' => 'failed', 'data' => null);
    sendJsonResponse($response);
}

function sendJsonResponse($sentArray)
{
    header('Content-Type: application/json');
    echo json_encode($sentArray);
}
```

```
}
```

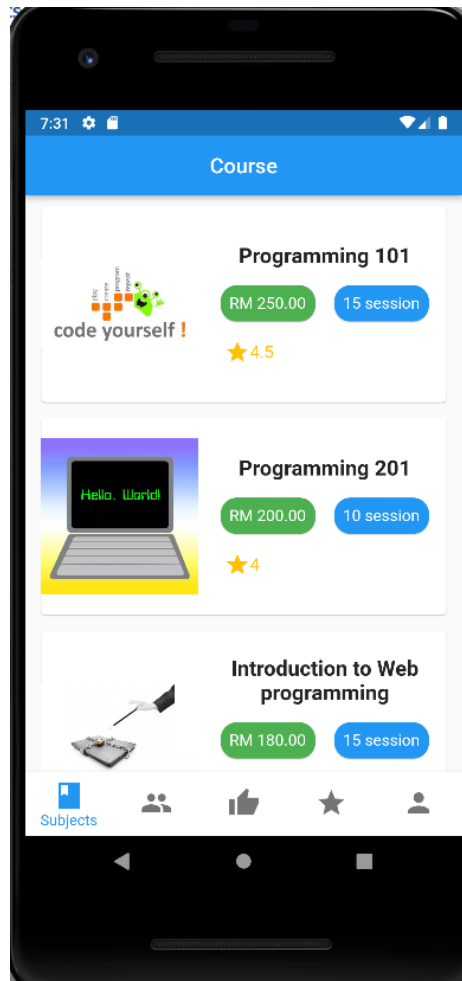
- iii. The Flutter method that implements the HTTP object POST call to login and creates a user object.

10 Marks

```
void _validateUser() {
  String email = _emailController.text;
  String password = _passwordController.text;
  http.post(Uri.parse("http://10.19.48.148/myTutorAPI/login.php"),
    body: {"email": email, "password": password}).then((response) {
    var data = jsonDecode(response.body);
    if (response.statusCode == 200 && data['status'] == 'success') {
      gb.globaUser = User.fromJson(data['data']);
      Fluttertoast.showToast(
        msg: "Success",
        toastLength: Toast.LENGTH_SHORT,
        gravity: ToastGravity.BOTTOM,
        timeInSecForIosWeb: 1,
        fontSize: 16.0);
      Navigator.of(context)
        .push(MaterialPageRoute(builder: (context) => HomeScreen()));
    } else {
      Fluttertoast.showToast(
        msg: data['status'],
        toastLength: Toast.LENGTH_SHORT,
        gravity: ToastGravity.BOTTOM,
        timeInSecForIosWeb: 1,
        fontSize: 16.0);
    }
  });
}
```

2. Main screen module: Once the user successfully login, navigate to the main page.
- a. The main page should implement a bottom navigation menu with the following menu icon; Subjects, Tutors, Subscribe, Favourite, and Profile. Show only the screenshot for the main page.

10 Marks



- b. Download the following SQL file and import it into your database. Show both imported tables screenshots using the PhpMyAdmin table designer. SQL - <https://slumberjer.com/mytutor/mytutordb.sql>

2 Marks

Tbl\_tutors

localhost / 127.0.0.1 / my\_tutor

Server: 127.0.0.1 Database: my\_tutor Table: tbl\_tutors

Showing rows 0 - 8 (9 total, Query took 0.0008 seconds)

```
SELECT * FROM `tbl_tutors`
```

Options

	tutor_id	tutor_email	tutor_phone	tutor_name	tutor_password	tutor_description	tutor_datereg
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	prashanthini@mytutor.com	+60188816970	Prashanthini a/l Manjit Ramasamy	6367c48dd193d56ea7b0baad25b19455e529f5ee	Nunc venenatis bibendum odio, in fermentum sem ut...	2022-06-02 10:29:14 000000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	sinnathuraybakzi@yang.com	+601132339126	Chai Tan Hui	86e40acd9d92d4c544de76963ab68208ef0ab389	Integer nulla dui, blandit eu sollicitudin vitae.	2022-05-24 15:21:25 000000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	huzir42@bakry.org	+6095798898	Nur Maya binti Aidil Hafizee	02dd38cccd56a4efbe224599f4717a8cf3eb9281	Maecenas vitae leo in ipsum pulvinar hendrerit vt...	2022-03-03 15:21:25 000000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	dkok@majid.biz	+60377236036	Ling Liang Thok	76b241504f3904db725c01fcc532c2bdfae609ae	Aliquam dignissim ut libero non aliquet. Etiam eu...	2022-06-09 21:47:05 000000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	melissa57@hotmail.com	+6069427992	Teoh Chum Liek	86e40acd9d92d4c544de76963ab68208ef0ab389	Morbi at turpis in quam gravida facilisis. Lorem ..	2022-06-27 21:48:25 000000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	azizy kavita@foong.info	+60152045292	Amirah binti Che Aznizam	02dd38cccd56a4efbe224599f4717a8cf3eb9281	Nullam sed fringilla nisi. Proin aliquet metus qui...	2022-03-03 15:21:25 000000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	leoki57@hotmail.com	+6069427992	Teoh Chum Liek	86e40acd9d92d4c544de76963ab68208ef0ab389	Morbi at turpis in quam gravida facilisis. Lorem ..	2022-06-27 21:48:25 000000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	nelson.jani@parthiban.info	+60102727980	Muhammet Firdaus Miskoulan bin	b4b9c99d312bf05bfb05341d981d5817ace24b51	Duis at iaculis ante. Praesent in natus blandit, ..	2022-05-29 22:10:53 000000

localhost / 127.0.0.1 / my\_tutor

Server: 127.0.0.1 Database: my\_tutor Table: tbl\_tutors

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	tutor_id	int(5)			No	None		AUTO_INCREMENT	<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More
2	tutor_email	varchar(50)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More
3	tutor_phone	varchar(15)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More
4	tutor_name	varchar(50)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More
5	tutor_password	varchar(40)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More
6	tutor_description	varchar(300)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More
7	tutor_datereg	datetime(6)			No	None			<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<input type="checkbox"/> Edit <input type="checkbox"/> Rename <input type="checkbox"/> Drop	PRIMARY	BTREE	Yes	No	tutor_id	9	A	No	
<input type="checkbox"/> Edit <input type="checkbox"/> Rename <input type="checkbox"/> Drop		BTREE	Yes	No	tutor_email	9	A	No	

Create an index on 1 column(s) after tutor\_datereg

Tbl\_subjects

Showing rows 0 - 14 (15 total. Query took 0.0009 seconds)

```
SELECT * FROM `tbl_subjects`
```

Options: Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	subject_id	subject_name	subject_description	subject_price	tutor_id	subject_sessions	subject_rating
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Programming 101	Basic programming for new student with no backgrou...	250	1	15	4.5
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Programming 201	Intermediate programming course that aim for those...	200	2	10	4
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	Introduction to Web programming	Basic introduction to HTML, CSS, JavaScript, PHP a...	180	5	15	3.8
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	Web programming advanced	Course aim for those who are familiar with basic w...	150	3	15	4.2
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	Python for Everybody	This Specialization builds on the success of the p...	120	6	5	4.5
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	Introduction to Computer Science	This specialisation covers topics ranging from bas...	180	3	10	4.2
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	Code Yourself! An Introduction to Programming	This course will teach you how to program in Scrat...	120	4	5	4
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	IBM Full Stack Software Developer Professional Cer...	ickstart your career in application development. M...	230	8	8	4.3
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	Graphic Design Specialization	his four-course sequence exposes students to the f...	200	1	5	4.2
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	Fundamentals of Graphic Design	At the end of this course you will have learned ho...	150	2	5	3.8
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	Fundamentals of Graphic Design	Project centered courses are designed specifically...	150	5	5	4.8

Table structure: Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	subject_id	int(4)			No	None		AUTO_INCREMENT	<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More
2	subject_name	varchar(150)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More
3	subject_description	varchar(500)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More
4	subject_price	float			No	None			<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More
5	tutor_id	varchar(4)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More
6	subject_sessions	int(3)			No	None			<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More
7	subject_rating	float			No	None			<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More

Indexes

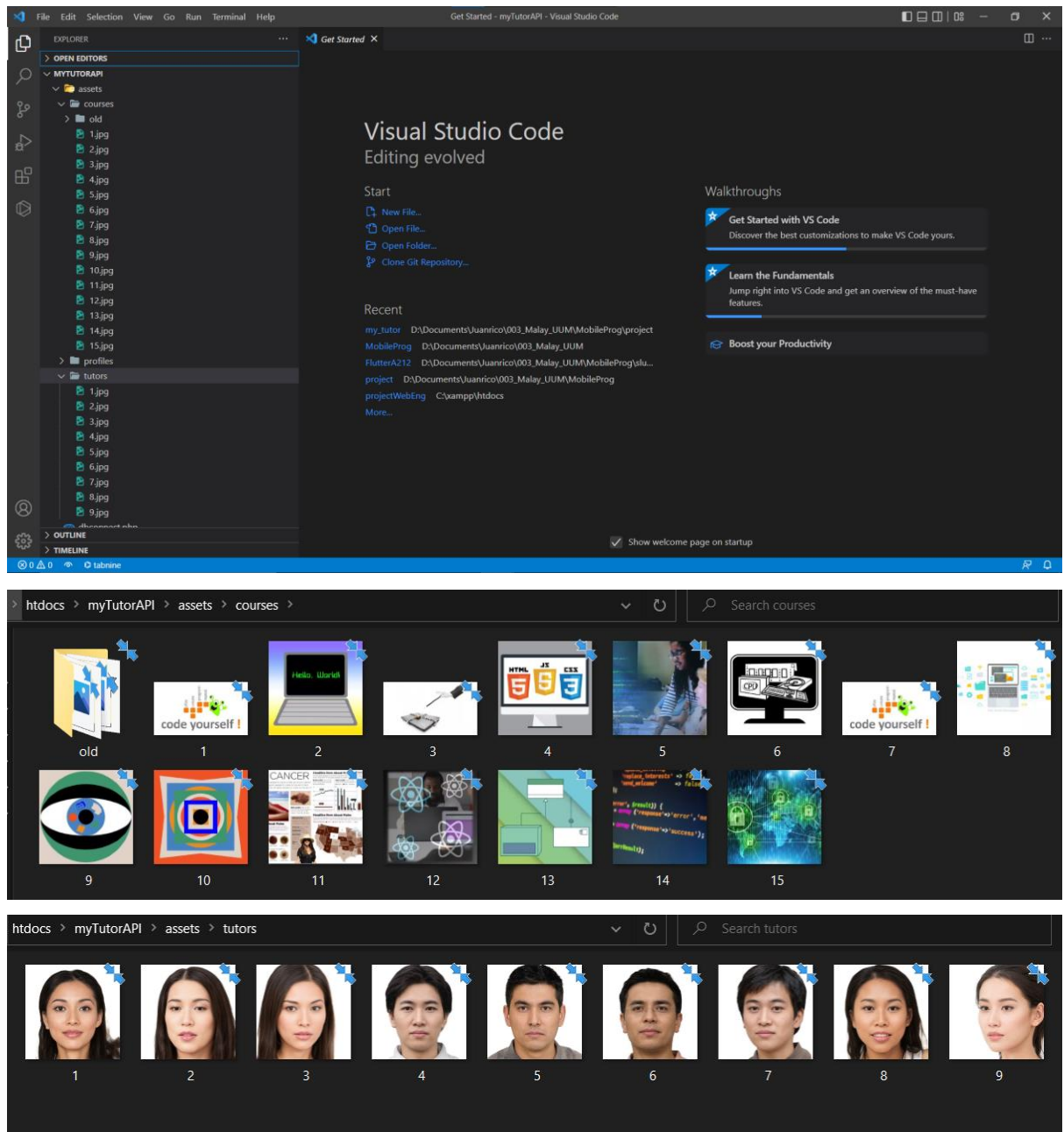
Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<input type="checkbox"/> Edit <input type="checkbox"/> Rename <input type="checkbox"/> Drop	PRIMARY	BTREE	Yes	No	subject_id	15	A	No	

Create an index on 1 column(s) Go

- c. Download the following assets files and extract the folder into your application directory. The folder contains two subfolders that store tutors' profile images and course images. Assets - <https://slumberjer.com/mytutor/assets.zip>. Print screen your server directory with all the assets.

2 Marks





- d. Once the user successfully login, the main page should display a list of subjects from the table `tbl_subjects` from the database. Load the required data from the database and display it as a card list. The image for each subject is provided in (2. c).

- i. Subject/Course class definition

5 Marks

```
class Course {
    String? subject_id;
    String? subject_name;
    String? subject_description;
    String? subject_price;
}
```



```
String? tutor_id;
String? subject_sessions;
String? subject_rating;

Course(
    {this.subject_id,
    this.subject_name,
    this.subject_description,
    this.subject_price,
    this.tutor_id,
    this.subject_sessions,
    this.subject_rating});

Course.fromJson(Map<String, dynamic> json) {
    subject_id = json['subject_id'].toString();
    subject_name = json['subject_name'];
    subject_description = json['subject_description'];
    subject_price = json['subject_price'].toString();
    tutor_id = json['tutor_id'];
    subject_sessions = json['subject_sessions'].toString();
    subject_rating = json['subject_rating'].toString();
}

Map<String, dynamic> toJson() {
    final Map<String, dynamic> data = <String, dynamic>{};
    data['subject_id'] = subject_id;
    data['subject_name'] = subject_name;
    data['subject_description'] = subject_description;
    data['subject_price'] = subject_price;
    data['tutor_id'] = tutor_id;
    data['subject_sessions'] = subject_sessions;
    data['subject_rating'] = subject_rating;
    return data;
}
}
```

- ii. PHP backend script with pagination that load subject/course list (5 items per page).

10 Marks

```
<?php
include("dbconnect.php");
session_start();
if (!isset($_POST)) {
    $response = array('status' => 'failed', 'data' => null);
```

```

        sendJsonResponse($response);
        die();
    }

    $limit = 5;
    $page = (isset($_POST['page']) && is_numeric($_POST['page'])) ? $_POST['page'] : 1;
    $paginationStart = ($page - 1) * $limit;
    $sqlSubject = "SELECT * FROM tbl_subjects LIMIT $paginationStart, $limit";
    $stmt = $conn->prepare($sqlSubject);
    $stmt->execute();
    $result = $stmt->get_result();

    // Get total records
    $sql = $conn->query("SELECT count(subject_id) AS id FROM tbl_subjects")->fetch_assoc();
    $allRecrods = $sql['id'];

    // Calculate total pages
    $totalPages = ceil($allRecrods / $limit);
    // Prev + Next
    $prev = $page - 1;
    $next = $page + 1;

    if ($allRecrods > 0) {
        $courses['courses'] = array();
        while ($row = $result->fetch_assoc()) {
            $clist = array();
            $clist['subject_id'] = $row['subject_id'];
            $clist['subject_name'] = $row['subject_name'];
            $clist['subject_description'] = $row['subject_description'];
            $clist['subject_sessions'] = $row['subject_sessions'];
            $clist['subject_rating'] = $row['subject_rating'];
            $clist['subject_price'] = $row['subject_price'];
            $clist['tutor_id'] = $row['tutor_id'];
            array_push($courses['courses'], $clist);
        }
        $response = array('status' => 'success', 'page' => "$page", 'totalPages' => "$totalPages", 'data' => $courses);
        sendJsonResponse($response);
    } else {
        $response = array('status' => 'failed', 'page' => "$page", 'totalPages' => "$totalPages", 'data' => null);
        sendJsonResponse($response);
    }
}

function sendJsonResponse($sentArray)
{

```

```

header('Content-Type: application/json');
echo json_encode($sentArray);
}
?>

```

- iii. The Flutter method that implements the HTTP object POST call to load the subject data into the list object.

10 Marks

I Use both tutor and subject data.

```

void loadCourse() {
  _numPages ?? 1;
  http.post(Uri.parse("http://10.19.48.148/myTutorAPI/load_subject.php"),
    body: {
      'page': _currentPage.toString(),
    }).timeout(
    const Duration(seconds: 5),
    onTimeout: () {
      return http.Response(
        'Error', 408); // Request Timeout response status code
    },
  ).then((response) {
    var jsondata = jsonDecode(response.body);
    if (response.statusCode == 200 && jsondata['status'] == 'success') {
      var extractdata = jsondata['data'];
      _numPages = int.parse(jsondata['totalPages']);

      if (extractdata['courses'] != null) {
        CourseList = <Course>[];
        extractdata['courses'].forEach((v) {
          CourseList.add(Course.fromJson(v));
        });
        titlecenter = CourseList.length.toString() + " Courses Available";
      } else {
        titlecenter = "No Course Available";
        CourseList.clear();
      }
      setState(() {});
    } else {
      titlecenter = "No Course Available 2";
      CourseList.clear();
      setState(() {});
    }
  });
}

```

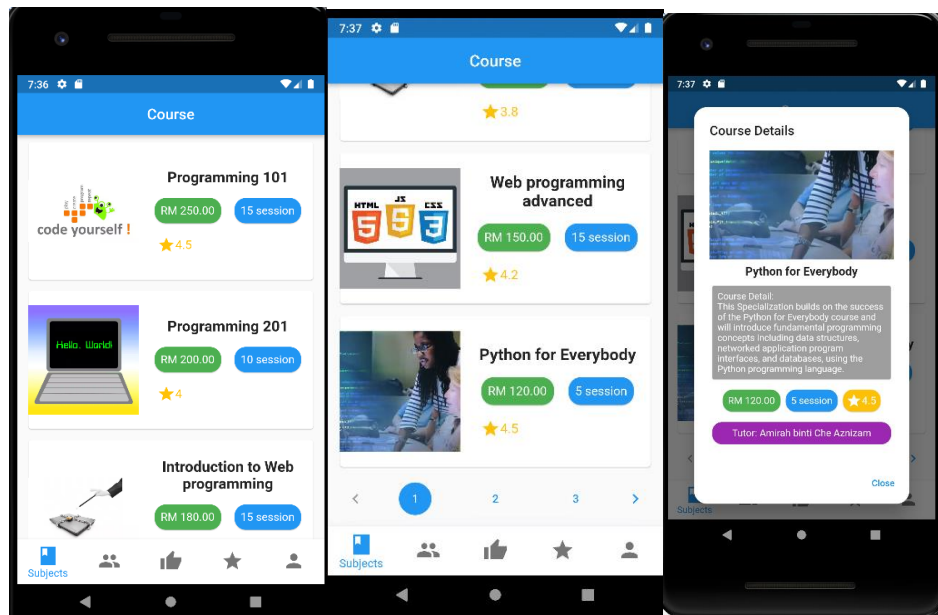
```
}

void loadTutor() {
    http.post(Uri.parse("http://10.19.48.148/myTutorAPI/load_tutor.php"),
        body: {
            'page': _currentPage.toString(),
            'limit': 'all',
        }).timeout(
        const Duration(seconds: 5),
        onTimeout: () {
            return http.Response(
                'Error', 408); // Request Timeout response status code
        },
    ).then((response) {
        var jsondata = jsonDecode(response.body);
        if (response.statusCode == 200 && jsondata['status'] == 'success') {
            var extractdata = jsondata['data'];

            if (extractdata['tutors'] != null) {
                TutorList = <Tutor>[];
                extractdata['tutors'].forEach((v) {
                    TutorList.add(Tutor.fromJson(v));
                });
            } else {
                TutorList.clear();
            }
            setState(() {});
        } else {
            TutorList.clear();
            setState(() {});
        }
    });
}
```

iv. Screenshot for the main page that shows a list of data.

2 Marks



3. Tutor List module. This module is accessible from the Tutor menu icon from the bottom navigation bar. Create a new class screen for listing available tutors from the tutor's table.

a. Tutor class definition

5 Marks

```
class Tutor {
    String? tutor_id;
    String? tutor_email;
    String? tutor_phone;
    String? tutor_name;
    String? tutor_description;
    String? tutor_datereg;

    Tutor(
        {this.tutor_id,
        this.tutor_email,
        this.tutor_phone,
        this.tutor_name,
        this.tutor_description,
        this.tutor_datereg});

    Tutor.fromJson(Map<String, dynamic> json) {
        tutor_id = json['tutor_id'].toString();
        tutor_email = json['tutor_email'].toString();
        tutor_phone = json['tutor_phone'].toString();
        tutor_name = json['tutor_name'].toString();
        tutor_description = json['tutor_description'].toString();
    }
}
```

```

    tutor_datereg = json['tutor_datereg'].toString();
}

Map<String, dynamic> toJson() {
    final Map<String, dynamic> data = <String, dynamic>{};
    data['tutor_id'] = tutor_id;
    data['tutor_email'] = tutor_email;
    data['tutor_phone'] = tutor_phone;
    data['tutor_name'] = tutor_name;
    data['tutor_description'] = tutor_description;
    data['tutor_datereg'] = tutor_datereg;
    return data;
}
}

```

- b. PHP backend script with pagination to load tutor list (5 items per page).

10 Marks

```

<?php
include("dbconnect.php");
session_start();
if (!isset($_POST)) {
    $response = array('status' => 'failed', 'data' => null);
    sendJsonResponse($response);
    die();
}

// Get total records
$sql = $conn->query("SELECT count(tutor_id) AS id FROM tbl_tutors")-
>fetch_assoc();
$allRecrods = $sql['id'];

$limit = (isset($_POST['limit'])) ? $allRecrods : 5;
$page = (isset($_POST['page']) && is_numeric($_POST['page'])) ? $_POST['page']
: 1;
$paginationStart = ($page - 1) * $limit;
$sqlTutor = "SELECT * FROM tbl_tutors LIMIT $paginationStart, $limit";
$stmt = $conn->prepare($sqlTutor);
$stmt->execute();
$result = $stmt->get_result();

// Calculate total pages
$totalPages = ceil($allRecrods / $limit);
// Prev + Next
$prev = $page - 1;

```

```

$next = $page + 1;

if ($allRecrods > 0) {
    $tutors['tutors'] = array();
    while ($row = $result->fetch_assoc()) {
        $tlist = array();
        $tlist['tutor_id'] = $row['tutor_id'];
        $tlist['tutor_email'] = $row['tutor_email'];
        $tlist['tutor_phone'] = $row['tutor_phone'];
        $tlist['tutor_name'] = $row['tutor_name'];
        $tlist['tutor_description'] = $row['tutor_description'];
        $tlist['tutor_datereg'] = $row['tutor_datereg'];
        array_push($tutors['tutors'], $tlist);
    }
    $response = array('status' => 'success', 'page' => "$page", 'totalPages' => "$totalPages", 'data' => $tutors);
    => sendJsonResponse($response);
} else {
    $response = array('status' => 'failed', 'page' => "$page", 'totalPages' => "$totalPages", 'data' => null);
    sendJsonResponse($response);
}

function sendJsonResponse($sentArray)
{
    header('Content-Type: application/json');
    echo json_encode($sentArray);
}
?>

```

- c. The Flutter method that implements the HTTP object POST call to load the subject data into the list object

10 Marks

```

void loadTutor() {
    _numPages ?? 1;
    http.post(Uri.parse("http://10.19.48.148/myTutorAPI/load_tutor.php"),
        body: {
            'page': _currentPage.toString(),
        }).timeout(
        const Duration(seconds: 5),
        onTimeout: () {
            return http.Response(
                'Error', 408); // Request Timeout response status code
        },
    ),

```



```

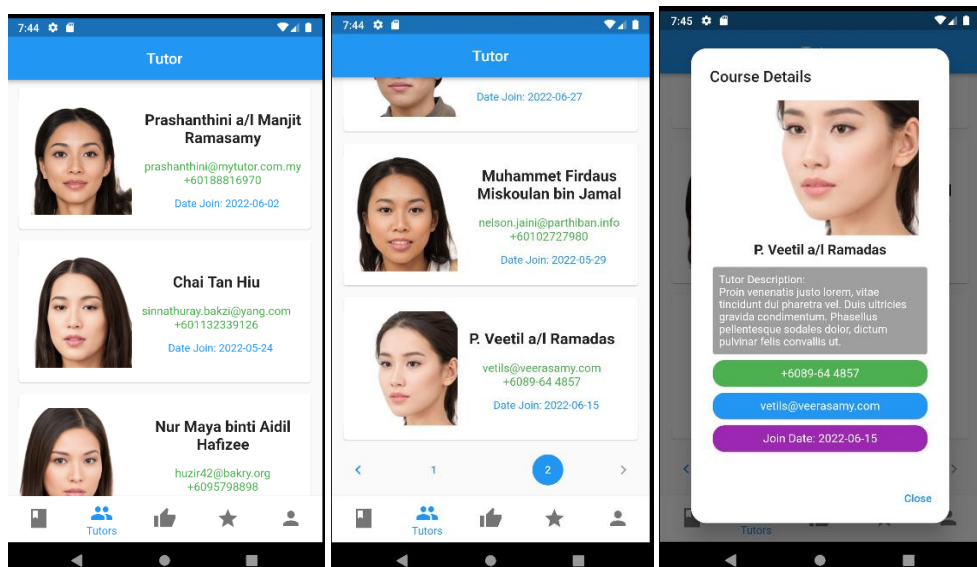
).then((response) {
    var jsondata = jsonDecode(response.body);
    if (response.statusCode == 200 && jsondata['status'] == 'success') {
        var extractdata = jsondata['data'];
        _numPages = int.parse(jsondata['totalPages']);

        if (extractdata['tutors'] != null) {
            TutorList = <Tutor>[];
            extractdata['tutors'].forEach((v) {
                TutorList.add(Tutor.fromJson(v));
            });
            // titlecenter = TutorList.length.toString() + " Tutors Available";
        } else {
            titlecenter = "No Tutor Available";
            TutorList.clear();
        }
        setState(() {});
    } else {
        titlecenter = "No Tutor Available 2";
        TutorList.clear();
        setState(() {});
    }
});
}
}

```

d. Screenshot for the main page that shows a list of data.

2 Marks



4. Create a presentation video that describes all the above tasks. The video must demonstrate the following:

**Put your video link here:** <https://youtu.be/6mW5gURV-7I>

a. Login screen

5 Marks

b. Subjects list screen (main screen)

5 Marks

c. Tutors list screen (tutor screen)

2 Marks