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

FRONT COVER

Lab 2

Name	Juanrico Alvaro
Matric No	702301
YouTube Presentation Link	https://youtu.be/oiHe1iIR_Wo
Phone Number	0136798361
GitHub Link	**https://github.com/victorico123/flutter-projrct-UUM**
Submission Date	22/5/2022
Acknowledgment	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.
Digital Signature	[illegible]
Students Picture	
Email use during book purchase	juanricoalvaro@gmail.com
Your digital key from the book purchase	5oRiMDDWKW4U5hz

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

Lab 2

Given date: 16/5/2022

Submission date: 23/5/2022

15 Marks

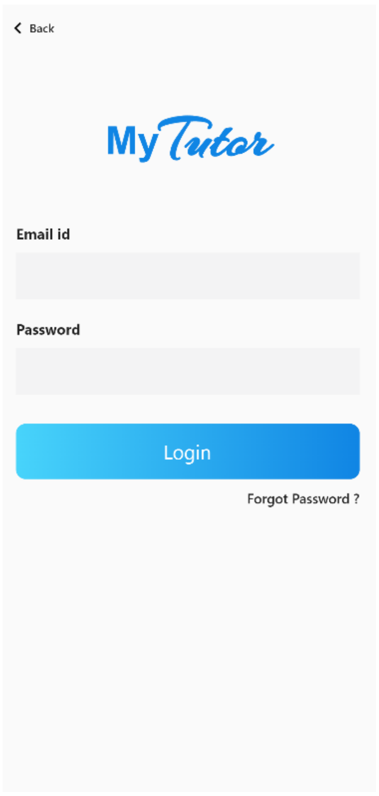
Create a new project and name **MY Tutor**. My tutor is a client-side application that use to search for an online tutor for a particular subject. Create a splash screen, user login screen, and user registration screen for the app. Create a database with a user table and implement a backend API service for user registration only. The registration page should include image uploads and other users' registration data such as email, name, phone number, password, and home address.

Answer the following questions. Don't show the entire code. Just select a segment of the codes responsible for the required questions. Make sure to retain the code format from VSC when you paste it into the Word document. Don't screenshot your code!.

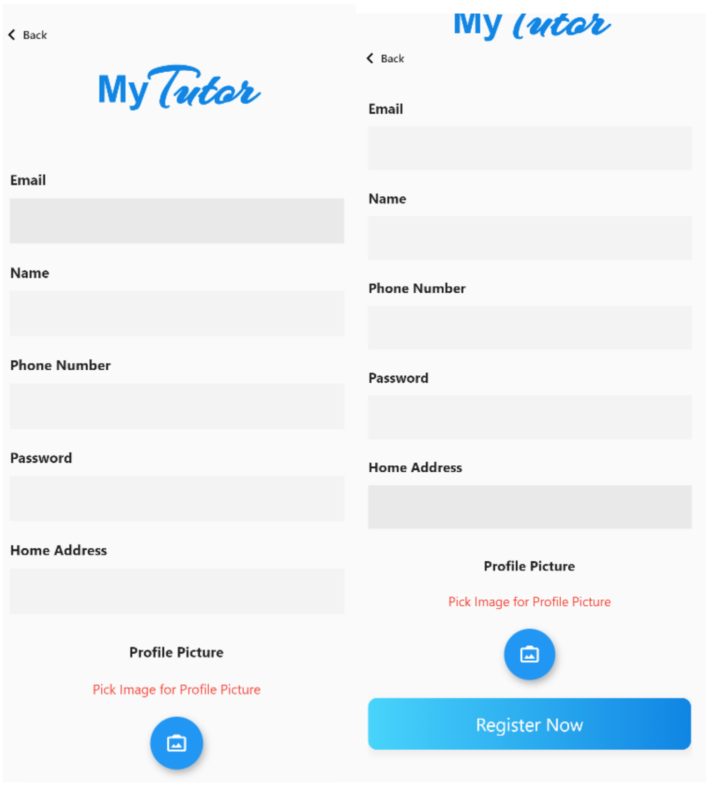
1. Show your user login and user registration user interface design for the application. You may use any digital tools that can help to build the user interfaces.

(2 Marks)

Login page



Register Page



2. Show code segment for the following tasks. (Your GitHub link to the project will be used as part of the evaluation):
   a. Build Widget for login and registration page.

(2 Marks)

Registration page

```
Widget build(BuildContext context) {
    final height = MediaQuery.of(context).size.height;
    return Scaffold(
      body: SizedBox(
        height: height,
        child: Stack(
          children: <Widget>[
            Container(
              padding: const EdgeInsets.symmetric(horizontal: 20),
              child: SingleChildScrollView(
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.center,
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: <Widget>[
                    SizedBox(height: height * .15),
                    _title(),
                    const SizedBox(
                      height: 40,
                    ),
                    _formWidget(),
                    const SizedBox(
                      height: 20,
                    ),
                    _uploadPictureWidget(),
                    const SizedBox(
                      height: 20,
                    ),
                    _submitButton(),
                    SizedBox(height: height * .05),
                  ],
                ),
              ),
            ),
            Positioned(top: 30, left: 0, child: _backButton()),
          ],
        ),
      ),
    );
  }
```

```
Widget _entryField(String title, TextEditingController myContoller,
      {bool isPassword = false}) {
    return Container(
      margin: const EdgeInsets.symmetric(vertical: 10),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: <Widget>[
        Text(
          title,
          style: const TextStyle(fontWeight: FontWeight.bold, fontSize: 15),
        ),
        const SizedBox(
          height: 10,
        ),
        TextFormField(
          controller: myContoller,
          obscureText: isPassword,
          keyboardType: title == "Phone Number"
              ? TextInputType.number
              : TextInputType.multiline,
          decoration: const InputDecoration(
              border: InputBorder.none,
              fillColor: Color(0xfff3f3f4),
              filled: true),
          validator: (value) {
            if (value == null || value.isEmpty) {
              return 'Please enter ' + title + ' field.';
            }
            return null;
          },
        )
      ],
    ),
  );
}

Widget _submitButton() {
  return GestureDetector(
    onTap: () => {_onSubmit()},
    child: Container(
      width: MediaQuery.of(context).size.width,
      padding: const EdgeInsets.symmetric(vertical: 15),
      alignment: Alignment.center,
      decoration: BoxDecoration(
          borderRadius: const BorderRadius.all(Radius.circular(10)),
          boxShadow: <BoxShadow>[
            BoxShadow(
                color: Colors.grey.shade200,
                offset: const Offset(2, 4),
                blurRadius: 5,
                spreadRadius: 2)
          ],
          gradient: const LinearGradient(
              begin: Alignment.centerLeft,
              end: Alignment.centerRight,
              colors: [
                Color.fromARGB(255, 72, 212, 251),
                Color.fromARGB(255, 16, 133, 228)
              ])),
      child: const Text(
        'Register Now',
        style: TextStyle(fontSize: 20, color: Colors.white),
      ),
    ),
  );
}

Widget _title() {
  return RichText(
    textAlign: TextAlign.center,
    text: TextSpan(
        text: 'My',
        style: const TextStyle(
            fontSize: 40,
            fontWeight: FontWeight.w700,
            color: Color.fromARGB(255, 16, 133, 228)),
        children: [
          TextSpan(
              text: 'Tutor',
              style: GoogleFonts.arizonia(
                  textStyle: const TextStyle(
                fontSize: 60,
                fontWeight: FontWeight.bold,
                color: Color.fromARGB(255, 16, 133, 228),
              )))
        ]),
  );
}

Widget _uploadPictureWidget() {
  return Center(
      child: Column(
    children: [
      const Text(
        "Profile Picture",
        style: TextStyle(fontWeight: FontWeight.bold, fontSize: 15),
      ),
      const SizedBox(
        height: 20,
      ),
      _image == null
          ? const Text("Pick Image for Profile Picture",
              style: TextStyle(color: Colors.red))
          : Image.file(_image),
      const SizedBox(
        height: 20,
      ),
      FloatingActionButton(
          onPressed: _galleryPicker,
          tooltip: "Pick Image for Profile Picture",
          child: const Icon(Icons.photo_camera_back))
    ],
  ));
}

_galleryPicker() async {
  final picker = ImagePicker();
  final pickedFile = await picker.pickImage(
    source: ImageSource.gallery,
    maxHeight: 200,
    maxWidth: 200,
  );
  if (pickedFile != null) {
    setState(() {
      _image = File(pickedFile.path);
      _imageExt = p.extension(pickedFile.path);
    });
  }
}

Widget _formWidget() {
  return Form(
    key: _formKey,
    child: Column(
      children: <Widget>[
        _entryField("Email", _emailController),
        _entryField("Name", _nameController),
        _entryField("Phone Number", _phoneController),
        _entryField("Password", _passwordController, isPassword: true),
        _entryField("Home Address", _addressController),
      ],
    ),
  );
}

_onSubmit() {
  if (_formKey.currentState!.validate() && _image != null) {
    _formKey.currentState!.save();
```

```
      _insertUser();
    }
  }
```

Login Page

```
Widget build(BuildContext context) {
    final height = MediaQuery.of(context).size.height;
    return Scaffold(
        body: SizedBox(
      height: height,
      child: Stack(
        children: <Widget>[
          Container(
            padding: const EdgeInsets.symmetric(horizontal: 20),
            child: SingleChildScrollView(
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.center,
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
                  SizedBox(height: height * .15),
                  _title(),
                  const SizedBox(height: 40),
                  _formWidget(),
                  const SizedBox(height: 20),
                  _submitButton(),
                  Container(
                    padding: const EdgeInsets.symmetric(vertical: 10),
                    alignment: Alignment.centerRight,
                    child: const Text('Forgot Password ?',
                        style: TextStyle(
                            fontSize: 14, fontWeight: FontWeight.w500)),
                  ),
                ],
              ),
            ),
          ),
          Positioned(top: 30, left: 0, child: _backButton()),
        ],
      ),
    ));
  }
```

```
Widget _backButton() {
    return InkWell(
      onTap: () {
        Navigator.pop(context);
      },
      child: Container(
        padding: const EdgeInsets.symmetric(horizontal: 10),
        child: Row(
          children: <Widget>[
            Container(
              padding: const EdgeInsets.only(left: 0, top: 10, bottom: 10),
              child: const Icon(Icons.keyboard_arrow_left, color:
Colors.black),
            ),
            const Text('Back',
                style: TextStyle(fontSize: 12, fontWeight: FontWeight.w500))
          ],
        ),
      ),
    );
  }

  Widget _entryField(String title, {bool isPassword = false}) {
    return Container(
      margin: const EdgeInsets.symmetric(vertical: 10),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: <Widget>[
          Text(
            title,
            style: const TextStyle(fontWeight: FontWeight.bold, fontSize: 15),
          ),
          const SizedBox(
            height: 10,
          ),
          TextField(
              obscureText: isPassword,
              decoration: const InputDecoration(
                  border: InputBorder.none,
                  fillColor: Color(0xfff3f3f4),
                  filled: true))
        ],
      ),
    );
  }

  Widget _submitButton() {
    return Container(
      width: MediaQuery.of(context).size.width,
      padding: const EdgeInsets.symmetric(vertical: 15),
      alignment: Alignment.center,
      decoration: const BoxDecoration(
    borderRadius: BorderRadius.all(Radius.circular(10)),
    gradient: LinearGradient(
      begin: Alignment.centerLeft,
      end: Alignment.centerRight,
      colors: [
       Color.fromARGB(255, 72, 212, 251),
       Color.fromARGB(255, 16, 133, 228)
      ])),
  child: const Text(
   'Login',
   style: TextStyle(fontSize: 20, color: Colors.white),
  ),
 );
}

Widget _title() {
 return RichText(
  textAlign: TextAlign.center,
  text: TextSpan(
    text: 'My',
    style: const TextStyle(
      fontSize: 40,
      fontWeight: FontWeight.w700,
      color: Color.fromARGB(255, 16, 133, 228)),
    children: [
     TextSpan(
       text: 'Tutor',
       style: GoogleFonts.arizonia(
         textStyle: const TextStyle(
        fontSize: 60,
        fontWeight: FontWeight.bold,
        color: Color.fromARGB(255, 16, 133, 228),
       )))
    ]),
 );
}

Widget _formWidget() {
 return Column(
  children: <Widget>[
   _entryField("Email id"),
   _entryField("Password", isPassword: true),
  ],
 );
}
```

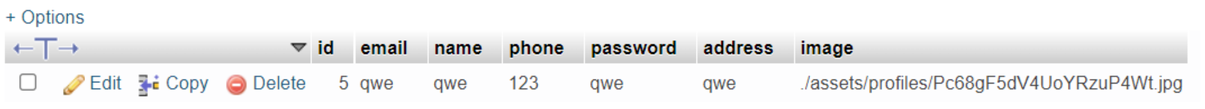
b. Method for user registration

(2 Marks)

```dart
void _insertUser() {
    String _email = _emailController.text;
    String _name = _nameController.text;
    String _phone = _phoneController.text;
    String _password = _passwordController.text;
    String _address = _addressController.text;
    String base64Image = base64Encode(_image!.readAsBytesSync());
    http.post(Uri.parse("http://10.19.105.124/myTutorAPI/register.php"), body: {
      "email": _email,
      "name": _name,
      "phone": _phone,
      "password": _password,
      "address": _address,
      "image": base64Image,
      "imageExt": _imageExt,
    }).then((response) {
      var data = jsonDecode(response.body);
      if (response.statusCode == 200 && data['status'] == 'success') {
        Fluttertoast.showToast(
            msg: "Success",
            toastLength: Toast.LENGTH_SHORT,
            gravity: ToastGravity.BOTTOM,
            timeInSecForIosWeb: 1,
            fontSize: 16.0);
        Navigator.of(context).pop();
      } else {
        Fluttertoast.showToast(
            msg: data['status'],
            toastLength: Toast.LENGTH_SHORT,
            gravity: ToastGravity.BOTTOM,
            timeInSecForIosWeb: 1,
            fontSize: 16.0);
      }
    });
  }
```

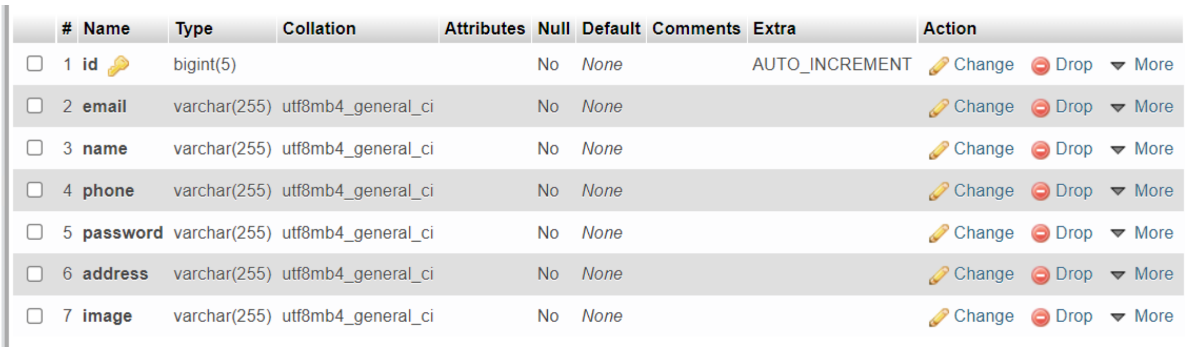
a. Database user table design screenshot.

(2 Marks)



+ Options

←T→				id	email	name	phone	password	address	image
☐	Edit	Copy	Delete	5	qwe	qwe	123	qwe	qwe	./assets/profiles/Pc68gF5dV4UoYRzuP4Wt.jpg



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(5)			No	None		AUTO_INCREMENT	Change Drop More
2	email	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	phone	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
5	password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
6	address	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
7	image	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

b. Backend PHP file for user registration.

(2 Marks)

dbConnect

```php
<?php
$servername = "localhost";
$username   = "root";
$password   = "";
$dbname     = "my_tutor";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}else{
    echo `<script>console.log('connection success');</script>`;
}
?>
```

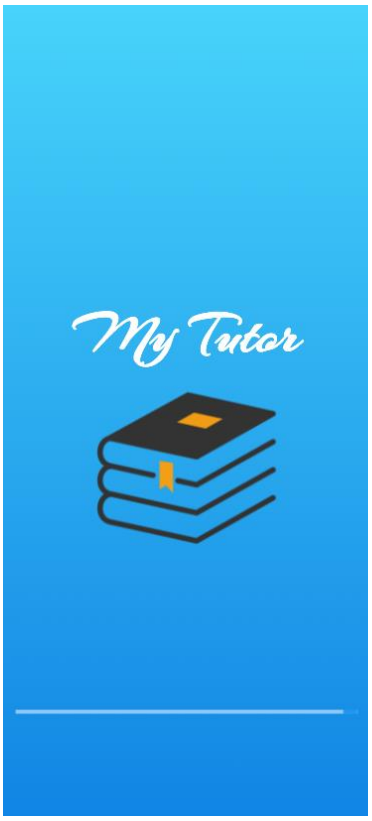
register backend

```php
<?php
if (!isset($_POST)) {
    $response = array('status' => 'failed', 'data' => null);
    sendJsonResponse($response);
    die();
}
include_once("dbconnect.php");
$email = addslashes($_POST['email']);
$name = addslashes($_POST['name']);
$phone = $_POST['phone'];
$password = $_POST['password'];
$address = addslashes($_POST['address']);
$base64image = $_POST['image'];
$imageExt = $_POST['imageExt'];
$tempFileName = getRandomString();
```

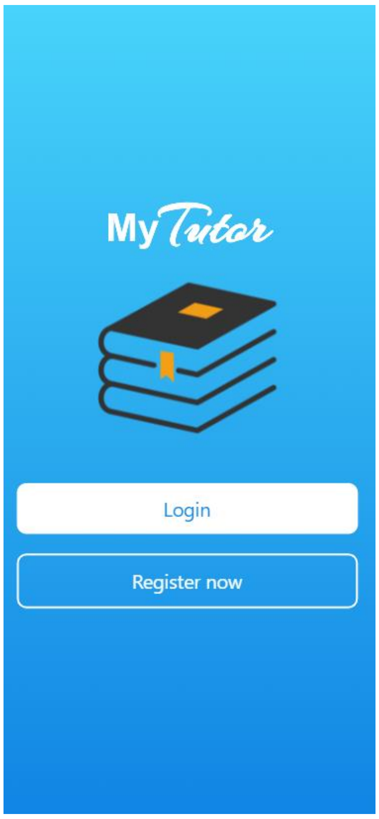
```php
$filePath = './assets/profiles/' . $tempFileName . $imageExt;
$decoded_string = base64_decode($base64image);
$is_written = file_put_contents($filePath, $decoded_string);
$sqlinsert = "INSERT INTO `users`(`email`, `name`, `phone`, `password`,
`address`,`image`) VALUES
('$email','$name','$phone','$password','$address','$filePath')";
if ($conn->query($sqlinsert) === TRUE) {
    $response = array('status' => 'success', 'data' => null);
    sendJsonResponse($response);
} else {
    $response = array('status' => 'failed', 'data' => null);
    sendJsonResponse($response);
}

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

function getRandomString() {
    $characters =
'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
    $randomString = '';

    for ($i = 0; $i < 20; $i++) {
        $index = rand(0, strlen($characters) - 1);
        $randomString .= $characters[$index];
    }

    return $randomString;
}
?>
```

3. Create and publish a YouTube presentation to demonstrate the registration process. Make sure to demonstrate the registration page successfully inserted a new user into the database. Show app screenshots for all pages (splash, login, registration).

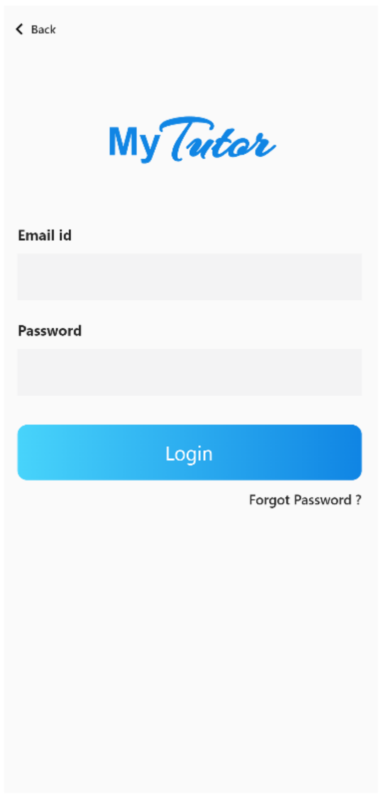
   Splash Screen



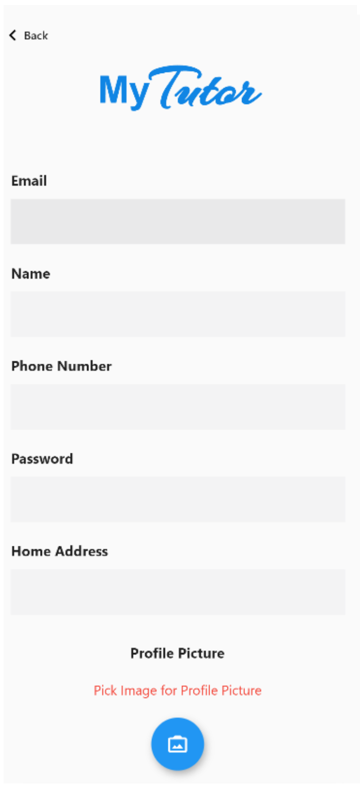
Landing Page

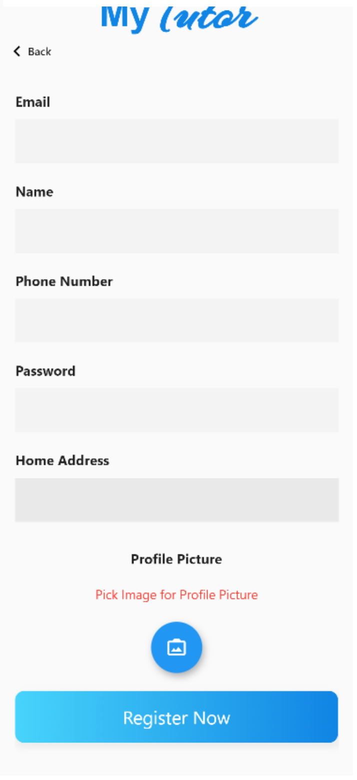


Login Page



Register Page





(5 Marks)

Submission

- This document with answers and filling all the required sections. Upload to learning when the link is open for submission.
- Create a short video for your app and upload it to YouTube (provide the link on the front page).
- Upload your project to your Github Repository (provide the link on the front page).