



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

FRONT COVER


Final Exam Assignment

Name	Juanrico Alvaro
Matric No	702301
YouTube Presentation Link	https://youtu.be/HvXxllktMHo
Phone Number	0136798361
GitHub Link	https://github.com/victorico123/flutter-projct-UUM
Submission Date	7/10/2022
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

Students Picture



	<p>STIW2044: Mobile Programming</p> <p>Semester A212</p> <p>School of Computing, CAS, UUM</p>
<p>Final Exam Assignment</p>	
<p>Given date: 3/7/2022</p>	
<p>Submission date: 7/7/2022</p>	
<p>100 Marks</p>	

A212 MOBILE PROGRAMMING FINAL (100 MARKS)

This final assignment is a continuation of your lab 3 of **MY Tutor** app.

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. Implement add to cart function from the course list screen. Create a button on each course card item and add the required information to the cart table. Based on the instruction, answer the following questions.

(25 Marks)

- a. Show the confirmation dialog method before adding a course to the cart.

5 Marks

```
void _confirmAdd(String index) {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        shape: const RoundedRectangleBorder(
          borderRadius: BorderRadius.all(Radius.circular(20.0))),
        title: const Text(
          "Course Details",
          style: TextStyle(),
        ),
        content: SingleChildScrollView(
          child: Center(
            child: Text("Are You sure to add this subject to your cart?"),
          ),
        ),
      );
    },
  );
}
```

```

    ),
    actions: [
      TextButton(
        child: const Text(
          "Cancel",
          style: TextStyle(),
        ),
        onPressed: () {
          Navigator.of(context).pop();
        },
      ),
      ElevatedButton(
        onPressed: () {
          addToCart(index);
          Navigator.of(context).pop();
        },
        child: Text("Add"))
    ],
  );
});
}

```

b. Show the method that implements the HTTP instruction to add a course to the cart.

10 Marks

```

void addToCart(String subjectId) {
  http.post(Uri.parse.gb.ip + "add_subject_to_cart.php"), body: {
    'subjectId': subjectId,
    'userId': gb.globaUser.id.toString(),
  }).timeout(
    const Duration(seconds: 5),
    onTimeout: () {
      return http.Response(
        'Error', 408); // Request Timeout response status code
    },
  ).then((response) {
    print(response.body);
    var jsondata = jsonDecode(response.body);
    if (response.statusCode == 200) {
      if (jsondata['status'] == 'success') {
        Fluttertoast.showToast(
          msg: "Success add to cart",
          toastLength: Toast.LENGTH_SHORT,
          gravity: ToastGravity.BOTTOM,
          timeInSecForIosWeb: 1,
          fontSize: 16.0);
      }
    }
  });
}

```

```

    } else {
      Fluttertoast.showToast(
        msg: "Item Already Exists",
        toastLength: Toast.LENGTH_SHORT,
        gravity: ToastGravity.BOTTOM,
        timeInSecForIosWeb: 1,
        fontSize: 16.0);
    }
  } else {
    Fluttertoast.showToast(
      msg: "Failed add to cart",
      toastLength: Toast.LENGTH_SHORT,
      gravity: ToastGravity.BOTTOM,
      timeInSecForIosWeb: 1,
      fontSize: 16.0);
  }
  Navigator.of(context).pop();
  setState(() {});
});
}

```

c. Show the backend script to handle 1.b.

10 Marks

```

<?php
if (!isset($_POST)) {
    $response = array('status' => 'failed', 'data' => null);
    sendJsonResponse($response);
    die();
}
include_once("dbconnect.php");
$subjectId = $_POST['subjectId'];
$userId = $_POST['userId'];
$sqlAdd = "SELECT * FROM tbl_carts WHERE `user_id` = ? AND `subject_id` = ?";
$stmt = $conn->prepare($sqlAdd);
$stmt->bind_param("ii", $subjectId, $userId);
if ($stmt->execute()) {
    $result = $stmt->get_result();
    if ($result->num_rows > 0) {
        $response = array('status' => 'registered', 'data' => null);
    } else {
        $sqlinsert = "INSERT INTO `tbl_carts`(`subject_id`, `user_id`) VALUES
('$subjectId','$userId')";
        if ($conn->query($sqlinsert) === TRUE) {
            $response = array('status' => 'success', 'data' => null);
        } else {

```

```

        $response = array('status' => 'failed', 'data' => null);
    }
}
sendJsonResponse($response);
} else {
    $response = array('status' => 'failed', 'data' => null);
    sendJsonResponse($response);
}

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

```

2. Create a new screen for listing cart content from the main screen. The cart content should list all the items added to the cart from 1. Based on the instruction, answer the following questions.

(30 Marks)

- a. Show the backend script that loads the cart list into an array object.

10 Marks

```

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

$userId = $_POST['userId'];
$sqlCart = "SELECT * FROM tbl_carts WHERE `user_id` = $userId";
$stmt = $conn->prepare($sqlCart);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    $courses['courses'] = array();
    while ($row = $result->fetch_assoc()) {
        $subjectId = $row['subject_id'];
    }
}

```

```

        $sqlSubject = "SELECT * FROM tbl_subjects WHERE `subject_id` =
$subjectId";
        $stmt2 = $conn->prepare($sqlSubject);
        $stmt2->execute();
        $result2 = $stmt2->get_result();
        while ($row2 = $result2->fetch_assoc()) {
            $clist = array();
            $clist['subject_id'] = $row2['subject_id'];
            $clist['subject_name'] = $row2['subject_name'];
            $clist['subject_description'] = $row2['subject_description'];
            $clist['subject_sessions'] = $row2['subject_sessions'];
            $clist['subject_rating'] = $row2['subject_rating'];
            $clist['subject_price'] = $row2['subject_price'];
            $clist['tutor_id'] = $row2['tutor_id'];
            array_push($courses['courses'], $clist);
        }
    }
    $response = array('status' => 'success', 'data' => $courses);
    sendJsonResponse($response);
} else {
    $response = array('status' => 'failed', 'data' => null);
    sendJsonResponse($response);
}

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

```

- b. Show the method that implements the HTTP instruction to load the requested data into the array list object.

10 Marks

```

void loadCartCourse() {
    http.post(Uri.parse(gb.ip + "load_cart.php"), body: {
        'userId': gb.globaUser.id.toString(),
    }).timeout(
        const Duration(seconds: 5),
        onTimeout: () {
            return http.Response(
                'Error', 408); // Request Timeout response status code
        },
    ).then((response) {
        print(response.body);
    });
}

```

```

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

  if (extractdata['courses'] != null) {
    CourseList = <Course>[];
    extractdata['courses'].forEach((v) {
      CourseList.add(Course.fromJson(v));
    });
    titlecenter = "Cart Available";
  } else {
    titlecenter = "Fail to get Course in Cart";
    CourseList.clear();
  }
} else {
  titlecenter = "No Course Available in cart";
  CourseList.clear();
}
setState(() {
  processTotalPayment();
});
});
}

void removeFromCart(String subjectId) {
  http.post(Uri.parse(gb.ip + "delete_from_cart.php"), body: {
    'subjectId': subjectId,
    'userId': gb.globaUser.id.toString(),
  }).timeout(
    const Duration(seconds: 5),
    onTimeout: () {
      return http.Response(
        'Error', 408); // Request Timeout response status code
    },
  ).then((response) {
    print(response.body);
    var jsondata = jsonDecode(response.body);
    if (response.statusCode == 200 && jsondata['status'] == 'success') {
      Fluttertoast.showToast(
        msg: "Success delete from cart",
        toastLength: Toast.LENGTH_SHORT,
        gravity: ToastGravity.BOTTOM,
        timeInSecForIosWeb: 1,
        fontSize: 16.0);
    } else {
      Fluttertoast.showToast(
        msg: "Failed delete from cart",
        toastLength: Toast.LENGTH_SHORT,

```

```
gravity: ToastGravity.BOTTOM,
timeInSecForIosWeb: 1,
fontSize: 16.0);
}
if (subjectId != "-1") {
  Navigator.of(context).pop();
}
setState(() {
  loadCartCourse();
  processTotalPayment();
});
});
}
```

c. Show only card widget tree (dart code) from Grid view.

5 Marks

5 Marks

[illegible]


```

        url) =>
const LinearProgressIndicator(),
  errorWidget: (context, url,
    error) =>
    const Icon(Icons.error),
  ),
),
),
Flexible(
  flex: 6,
  child: Column(
    mainAxisAlignment:
      MainAxisAlignment
        .center,
    children: [
      Container(
        margin: const EdgeInsets
          .all(5),
        child: Text(
          CourseList[index]
            .subject_name
              .toString(),
          style:
            const TextStyle(
              fontSize: 20,
              fontWeight:
                FontWeight.bold,
            ),
          textAlign:
            TextAlign.center,
        ),
      ),
      Container(
        margin: const EdgeInsets
          .all(8),
        child: Row(
          mainAxisAlignment:
            MainAxisAlignment
              .center,
          children: [
            Card(
              color:
                Colors.green,
              shape:
                RoundedRectangleBorder(
                  borderRadius:
                    BorderRadius
                      .circular(15.0),

```

```

    ),
    child: Container(
      margin:
        const EdgeInsets
          .all(8),
      child: Text(
        "RM " +
double.parse(CourseList[index].subject_price.toString())
        .toStringAsFixed(
          2),
        style:
          const TextStyle(
            color: Colors
              .white,
            fontSize:
              15,
          )),
    ),
    const SizedBox(
      width: 10,
    ),
    Card(
      color:
        Colors.blue,
      shape:
        RoundedRectangleBorder(
          borderRadius:
            BorderRadius
              .circular(
                15.0),
        ),
      child: Container(
        margin:
          const EdgeInsets
            .all(8),
        child: Text(
          CourseList[index]
            .subject_sessions
              .toString() +
            " session",
          style:
            const TextStyle(
              color: Colors
                .white,
              fontSize:
                15,
            )),

```

```

        ),
      ),
    ],
  ),
),
Container(
  alignment: Alignment
    .centerLeft,
  margin:
    const EdgeInsets
      .fromLTRB(
        20, 5, 5, 5),
  child: Row(
    mainAxisAlignment:
      MainAxisAlignment
        .min,
    children: [
      const Icon(
        Icons.star,
        color: Colors
          .amber),
      Text(
        CourseList[
          index]
          .subject_rating
            .toString(),
        style:
          const TextStyle(
            color: Colors
              .amber,
            fontSize:
              16,
          )),
    ],
  )),
  ],
  )),
),
);
},
),
Padding(
  padding: const EdgeInsets.all(10),
  child: Center(
    child: Text(
      "Total Payment : " +

```

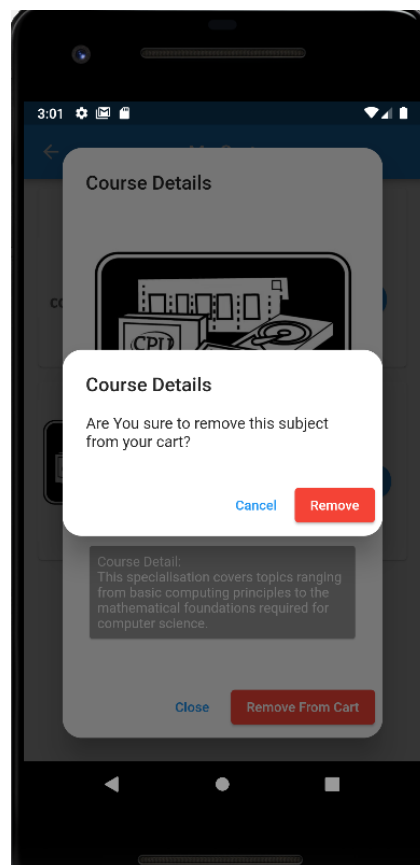
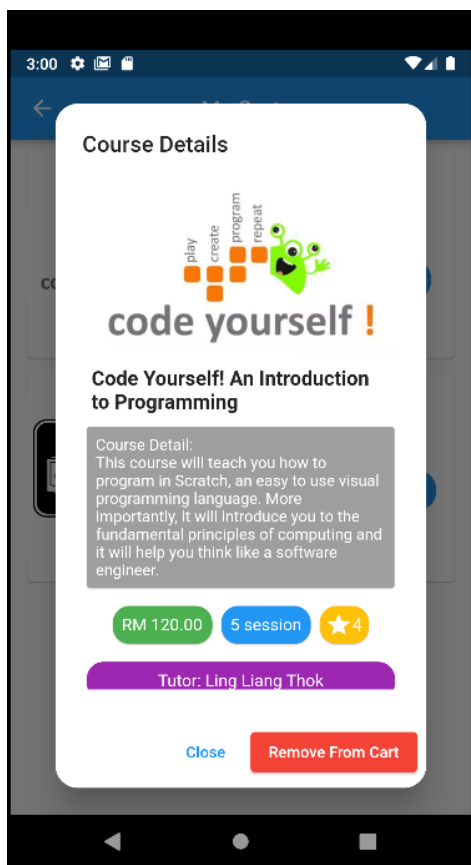
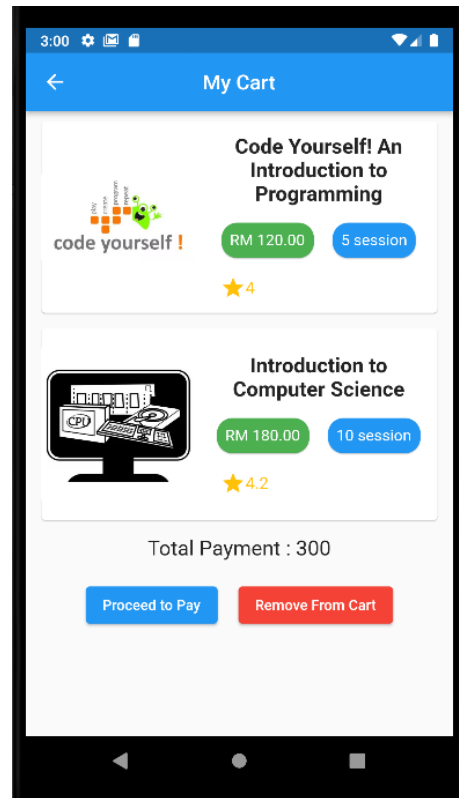
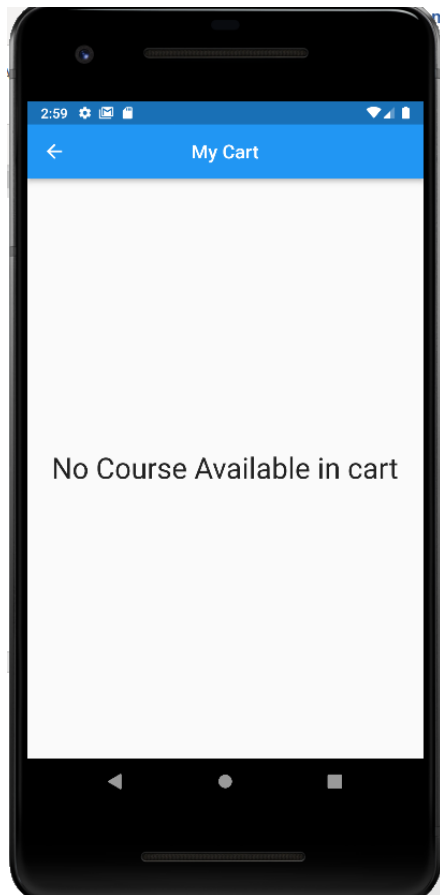
```

        totalPay.toString(),
        style:
            const TextStyle(fontSize: 20))),
    Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Padding(
          padding: const EdgeInsets.all(10),
          child: ElevatedButton(
            onPressed: _payment,
            child: Text('Proceed to Pay'))),
        Padding(
          padding: const EdgeInsets.all(10),
          child: ElevatedButton(
            style: ElevatedButton.styleFrom(
              primary: Colors.red, // background
              onPrimary:
                Colors.white, // foreground
            ),
            onPressed: () {
              _removeCart("-1");
            },
            child: Text("Remove From Cart"))),
      ],
    ),

```

d. Show screenshots for the cart screen.

5 Marks



3. The cart screen will require implementing the payment module. There are two options for the payment method either through Billplz payment gateway sandbox (or other payment gateway sandbox). Sandbox payment gateway will require a live Webhosting (e.g. free <https://www.infinityfree.net/> | <https://www.000webhost.com/> or paid (<https://system.jomhosting.net/aff.php?aff=2455>)). Answer the following questions based on the instruction above.

(30 Marks)

- a. Create a payment screen to handle the payment process. Show the web view widget code.

5 Marks

```
WebView(  
    initialUrl: gb.ip +  
        'payment.php?email=' +  
        gb.globaUser.email.toString() +  
        '&mobile=' +  
        gb.globaUser.phone.toString() +  
        '&name=' +  
        gb.globaUser.name.toString() +  
        '&amount=' +  
        widget.totalPay.toString(),  
    javascriptMode: JavascriptMode.unrestricted,  
    onWebViewCreated: (WebViewController webViewController) {  
        _controller.complete(webViewController);  
    },  
),
```

- b. Implement backend script handling the payment system. Show the required script to handle the payment.

5 Marks

```
<?php  
  
$email = $_GET['email'];  
$mobile = $_GET['mobile'];  
$name = $_GET['name'];  
$amount = $_GET['amount'];  
  
$api_key = '35284b93-1247-46f3-83d1-50d1893ce274';  
$collection_id = 'gv017jyu';  
  
$host = 'https://www.billplz-sandbox.com/api/v3/bills';
```

```

$data = array(
    'collection_id' => $collection_id,
    'email' => $email,
    'mobile' => $mobile,
    'name' => $name,
    'amount' => ($amount + 1) * 100, // RM20
    'description' => 'Payment for order by '.$name,
    'callback_url' => "http://victorico.id/mytutor/return_url",
    'redirect_url' =>
"http://victorico.id/mytutor/payment_update.php?email=$email&mobile=$mobile&am
ount=$amount&name=$name"
);

$process = curl_init($host );
curl_setopt($process, CURLOPT_HEADER, 0);
curl_setopt($process, CURLOPT_USERPWD, $api_key . ":");
curl_setopt($process, CURLOPT_TIMEOUT, 30);
curl_setopt($process, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($process, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($process, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt($process, CURLOPT_POSTFIELDS, http_build_query($data) );

$return = curl_exec($process);
curl_close($process);

$bill = json_decode($return, true);
var_dump($bill);
header("Location: {$bill['url']}");
?>

```

- c. Handle the backend for success or failed payment operation. Update relevant database tables. Show only the backend section where the database update happens.

10 Marks

```

<?php
error_reporting(0);
include_once("dbconnect.php");
$email = $_GET['email'];
$amount = $_GET['amount'];
$name = $_GET['name'];

$data = array(
    'id' => $_GET['billplz']['id'],
    'paid_at' => $_GET['billplz']['paid_at'] ,
    'paid' => $_GET['billplz']['paid'],

```

```

        'x_signature' => $_GET['billplz']['x_signature']
    );

    $paidstatus = $_GET['billplz']['paid'];
    if ($paidstatus=="true"){
        $paidstatus = "Success";
        $status = "Paid";
    }else{
        $paidstatus = "Failed";
        $status = "Failed";
    }
    $receiptid = $_GET['billplz']['id'];
    $signing = '';
    foreach ($data as $key => $value) {
        $signing.= 'billplz'.$key . $value;
        if ($key === 'paid') {
            break;
        } else {
            $signing .= '|';
        }
    }
}

$message = "hahah";
$error = 1;
$userid = 0;
$signed= hash_hmac('sha256', $signing, 'S-m1UsqkozdqhjD5tRSqv5SA');
if ($signed === $data['x_signature']) {
    if ($paidstatus == "Success"){ //payment success the back button on the
app
        $sqlGetUserId = "SELECT * FROM `users` WHERE `email` = '$email'";
        if ($result1 = $conn->query($sqlGetUserId)) {
            if ($result1->num_rows > 0) {
                while ($rows1 = $result1->fetch_assoc()) {
                    $userid = $rows1['id'];
                }
                $message = "-1";
            }
            $sqlGetCart = "SELECT * FROM `tbl_carts` WHERE
`user_id`='$userid'";
            $message = "1";
            if ($result2 = $conn->query($sqlGetCart)) {
                if ($result2->num_rows > 0) {
                    while ($rows2 = $result2->fetch_assoc()) {
                        $subjectid = $rows2['subject_id'];
                        $sqlInsertHistory = "INSERT INTO `tbl_history`(
`receipt_id`,`user_id`,`subject_id`) VALUES
('$receiptid','$userid','$subjectid')";

```



```

        $conn->query($sqlInsertHistory);
    }
    $message = "2";
    $sqlInsertPayment = "INSERT INTO `tbl_payment` (
`receipt_id`,`user_id`,`payment_total_paid`) VALUES
('$receiptid','$userid','$amount')";
    $sqlDeleteCart = "DELETE FROM `tbl_carts` WHERE
`user_id`='$userid'";

    if ($conn->query($sqlInsertPayment) && $conn-
>query($sqlDeleteCart)){
        $error= 0;
    }
}
}

}

}
if($error==1){

    // $message = "Payment FAILED.";
    printTable('Failed',$name,$email,$amount,$paidstatus,$message);
}else{
    $message = "Payment COMPLETE. ";
    $amount = number_format((float)$amount, 2, '.', '');
    printTable($receiptid,$name,$email,$amount,$paidstatus,$messag
e);
}

}

function printTable($receiptid,$name,$email,$amount,$paidstatus,$message){
    echo "
    <html>
    <head>
        <meta name='viewport' content='width=device-width, initial-
scale=1'>
        <link rel='stylesheet'
href='https://www.w3schools.com/w3css/4/w3.css'>
    </head>
    <div class='w3-padding'> <h4>Thank you for your payment</h4>
    <p>The following is your receipt for recent payment</p></div>
    <div class='w3-container w3-padding'>

```

```

        <table class='w3-table w3-striped w3-bordered'>
        <tr><th>Receipt ID</th><td>$receiptid</td></tr>
        <tr><th>Paid By</th><td>$name</td></tr>
        <tr><th>Email</th><td>$email</td></tr>
        <tr><th>Amount </th><td>RM $amount</td></tr>
        <tr><th>Payment Status</th><td>$paidstatus</td></tr>
        </table>
    <hr>";
    if($receiptid == 'Failed'){
        echo "<div class='w3-container w3-round w3-block w3-
red'><b>$message</b></div>";
    }else{
        echo "<div class='w3-container w3-round w3-block w3-
green'><b>$message</b></div>";

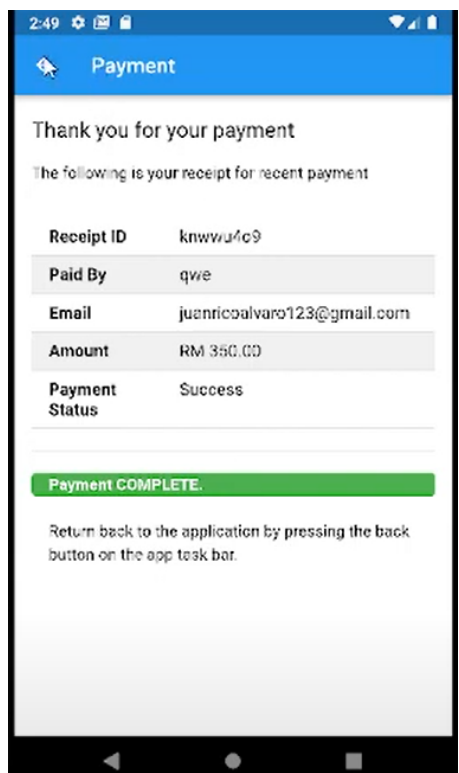
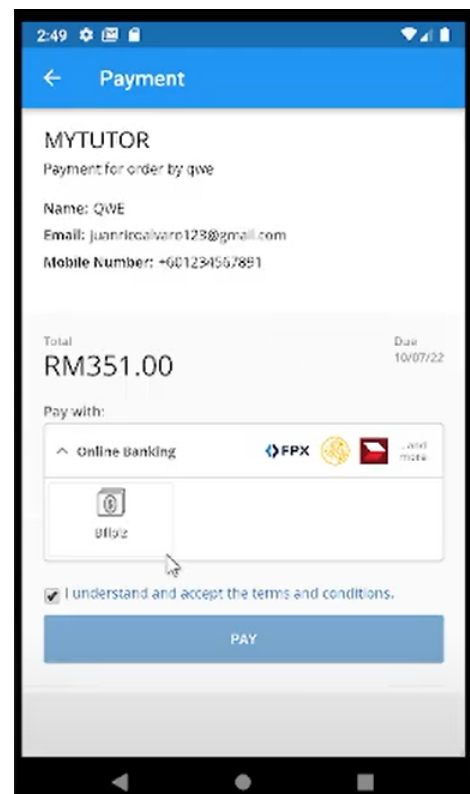
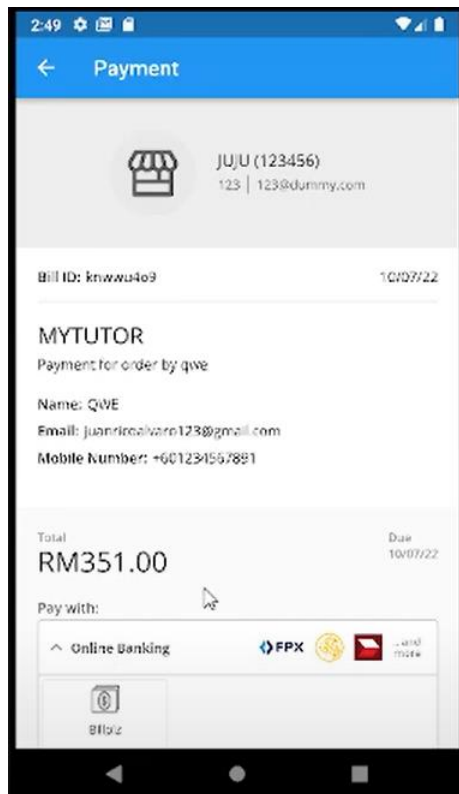
    }

    echo "<br>
    <div class='w3-container w3-round w3-block'>Return back to the
application by pressing the back button on the app task bar.</div>
    </div>
    </body></html>
    ";
}
?>

```

d. Show relevant print screens from the application for the payment process.

5 Marks



- e. Show the dashboard from the payment gateway application indicating the successful payment.

5 Marks

Billplz

Juju

DASHBOARD

BILLING

PAYMENT FORM

PAYOUT API

API

About

FAQ

Terms

Policy

Support

Credit balance: RM9,984.00

This account is active until 09/07/23 Sign out

Billing MYTUTOR

Collection ID: gv017jyu

Copy Collection ID | Payment Method | Split Rule | Email Notification

All Due Paid Deleted

Q ID Search input...

Delete All Add Bill

ID	DATE	NAME	EMAIL	MOBILE NUMBER	TOTAL (RM)	STATUS	ACTIONS
knwueu4o9	10/07/2022	QWE	juanricoalvaro123@gmail.com	+601234567891	351.00	Paid 10/07/2022	View receipt
vya2uyea	10/07/2022	QWE	juanricoalvaro123@gmail.com	+601234567891	201.00	Paid 10/07/2022	View receipt
ia8gkhgh	10/07/2022	QWE	juanricoalvaro123@gmail.com	+601234567891	251.00	Paid 10/07/2022	View receipt
mj9qgeix	10/07/2022	QWE	juanricoalvaro123@gmail.com	+601234567891	251.00	Paid 10/07/2022	View receipt
y207mlgk	10/07/2022	QWE	juanricoalvaro123@gmail.com	+601234567891	251.00	Due 10/07/2022	View bill
jd35w0k0	10/07/2022	QWE	juanricoalvaro123@gmail.com	+601234567891	251.00	Paid 10/07/2022	View receipt
hccmotpr	10/07/2022	QWE	juanricoalvaro123@gmail.com	+601234567891	251.00	Paid	View receipt

4. Create a presentation video that shows 1,2 and 3 modules.

(15 Marks)

<https://youtu.be/HvXxllktMHo>

- Add to cart
- Cart Content
- Payment