

Answer of Assessment Hard Skill

Note

this repo is just to store my answer from the assessment hard skill

Level 1-2

Structure Query Language (SQL) - Basic (HS)

1. Create the Students table.

```
CREATE TABLE Students (  
  student_id INT PRIMARY KEY,  
  first_name TEXT,  
  last_name TEXT,  
  age INT,  
  grade TEXT  
);
```



2. Retrieve All Student

```
SELECT * FROM Students;
```



Mobile Programming (Android/IOS) - Basic (HS)

I choose React Native Framework to create Mobile Programming task due to the programming language that I've learned and I have experienced with it.

```
import React from 'react';  
import { View, Text, StyleSheet } from 'react-native';  
  
const App = () => {  
  return (  
    <View style={styles.container}>  
      <Text style={styles.text}>Hello, Mobile World!</Text>  
    </View>  
  );  
};
```



```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#f0f0f0',
  },
  text: {
    fontSize: 24,
    fontWeight: 'bold',
    color: '#333',
  },
});

export default App;
```

To execute this code and run the app:

- Set up development environment by installing Node.js, npm, and React Native CLI.
- Create a new React Native project using the `react-native init MyApp` command in terminal.
- Replace the default contents of the `App.js` or `index.js` file in the newly created React Native project with the code provided above.
- Run the application on a simulator/emulator or a physical device using the appropriate command (`react-native run-android` for **Android** or `react-native run-ios` for **iOS**).

Frontend Programming (Html/ CSS/Bootstrap/CI) - Basic (HS)

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>My Web Page</title>
</head>

<body>
  <h1>Welcome to My Web Page</h1>
</body>

</html>
```



Back-end Programming (PHP/ C#/C++/Dart) - Basic (HS)

1. Node.js server

```
const express = require('express');

const app = express();

app.get('/hello', (req, res) => {
  res.json({ message: 'Hello, World!' });
});

const PORT = 3000;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```



2. SQL

```
SELECT * FROM users;
```



Level 3

Structure Query Language (SQL) - Intermediate / Internal Training (HS)

1. Subquery

```
SELECT c.name AS customer_name, o.order_date
FROM Customers c
INNER JOIN Orders o ON c.customer_id = o.customer_id
WHERE o.quantity > (
  SELECT AVG(quantity)
  FROM Orders
)
```



the result would look like this:

customer_name	order_date
John Doe	2023-01-10

2. JOIN Operation

```
SELECT c.name AS customer_name, o.product
FROM Customers c
JOIN Orders o ON c.customer_id = o.customer_id;
```



the result would look like this:

customer_name	product
John Doe	Laptop
Jane Smith	Smartphone
John Doe	Tablet

Data Visualization using Business Intelligence (Fine Bi or Other) - Intermediate / Internal Training (HS)

FineBI

Mobile Programming (Android/ IOS) - Intermediate / Internal Training (HS)

```
import React from 'react';
import { StyleSheet, Text, View } from 'react-native';

export default function App() {
  return (
    <View style={styles.container}>
      <Text style={styles.text}>Hello, Android!</Text>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#fff',
  },
  text: {
    fontSize: 24,
```



```
    fontWeight: 'bold',  
    color: '#333',  
  },  
});
```

Frontend Programming (Html/ CSS/ Bootstrap/ CI) - Intermediate/ Internal Training (HS)

```
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
  <meta charset="UTF-8">  
  <title>My Personal Profile - AdiSite</title>  
  <style>  
    * {  
      margin: 0;  
      padding: 0;  
      box-sizing: border-box;  
    }  
  
    .contact-form {  
      width: 300px;  
      margin: 20px auto;  
      padding: 20px;  
      border: 1px solid #ccc;  
      border-radius: 5px;  
      background-color: #f9f9f9;  
    }  
  
    .form-group {  
      margin-bottom: 15px;  
    }  
  
    label {  
      display: block;  
      margin-bottom: 5px;  
      font-weight: bold;  
    }  
  
    input[type="text"],  
    input[type="email"],  
    textarea {  
      width: 100%;  
      padding: 8px;  
      border: 1px solid #ccc;
```



```
        border-radius: 3px;
        font-size: 14px;
    }

    button[type="submit"] {
        width: 100%;
        padding: 10px;
        border: none;
        border-radius: 3px;
        background-color: #007bff;
        color: #fff;
        font-size: 16px;
        cursor: pointer;
    }

    button[type="submit"]:hover {
        background-color: #0056b3;
    }
</style>

</head>

<body>
    <h1>Personal Profile:</h1>

    <form action="#" method="post" class="contact-form">
        <div class="form-group">
            <label for="name">Name:</label>
            <input type="text" id="name" name="name" placeholder="ex: adi g"
required>
        </div>

        <div class="form-group">
            <label for="email">Email:</label>
            <input type="email" id="email" name="email" placeholder="Enter your
email" required>
        </div>

        <div class="form-group">
            <label for="comments">Comments:</label>
            <textarea id="comments" name="comments" placeholder="Enter your
comments, bad word is allowed" rows="5" required></textarea>
        </div>

        <button type="submit">Submit</button>
    </form>
</body>
```

```
</html>
```

Back-end Programming (PHP/ C#/ C++/ Dart) - Intermediate / Internal Training (HS)

Session destroy is: to terminate or clear a session, the example of implementation session destroy is at logout function.

```
<?php
session_start();

if (isset($_SESSION['user'])) {
    unset($_SESSION['user']);

    // Destroy the session
    session_destroy();

    echo 'Session destroyed. User logged out.';
} else {
    echo 'No active session found.';
}
?>
```



Level 4

Structure Query Language (SQL) - Expert / External Certifications (HS)

SQL injection is a type of security vulnerability that occurs when an attacker can manipulate queries by injecting malicious code into input fields of an application.

Security strategies or techniques that I ever implemented to prevent this security issue in PHP is using function `mysqli_real_escape_string()`, here an example:

```
<?php
$username = mysqli_real_escape_string($connection, $_POST['username']);
$sql = "SELECT * FROM users WHERE username = '$username'";
?>
```



Data Visualization using Business Intelligence (Fine Bi or Other) - Expert / External Certifications (HS)

FineBI

Mobile Programming (Android/ IOS) - Expert / External Certifications (HS)

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- ViewPager to display fragments -->
    <androidx.viewpager.widget.ViewPager
        android:id="@+id/viewPager"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</RelativeLayout>
```



fragment_one.xml

```
<!-- fragment_one.xml -->
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fragment One"
        android:textSize="20sp"
        android:textStyle="bold" />

    <!-- Add other views or components as needed for Fragment One -->

</LinearLayout>
```



fragment_two.xml

```
<!-- fragment_two.xml -->
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fragment Two"
        android:textSize="20sp"
        android:textStyle="bold" />

    <!-- Add other views or components as needed for Fragment Two -->

</LinearLayout>
```



fragment_three.xml

```
<!-- fragment_three.xml -->
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fragment Three"
        android:textSize="20sp"
        android:textStyle="bold" />

    <!-- Add other views or components as needed for Fragment Three -->

</LinearLayout>
```



Frontend Programming (Html/ CSS/ Bootstrap/ CI) - Expert / External Certifications (HS)

To ensure cross browser compatibility firstly make sure we Follow the latest HTML, CSS, and JavaScript standards recommended by the W3C , include `<!DOCTYPE html>` and `<meta name="viewport" content="width=device-width, initial-scale=1.0" />` HTML code. and here another step that is also important:

- use CSS Reset,
- do a test on multiple Browsers and Devices,
- use appropriate vendor prefixes (`-webkit-`, `-moz-`, `-ms-`, `-o-`) for CSS properties.
- use media queries in css like min-widht and max-widht
- use relative unit of measurement for component that need to follow screen size / responsive like (`rem`, `percentage`, `em`) all of I've mentioned can address difference appearance or behavior across various browser

Back-end Programming (PHP/ C#/ C++/ Dart) - Expert / External Certifications (HS)

Optimizing the performance of a slow-running PHP script processing a large amount of data involves identifying bottlenecks and implementing improvements in code efficiency, database interactions, and server configurations. Here are several techniques and tools to enhance PHP code performance:

- Code Profiling

Use PHP profiling tools like Xdebug or built-in functions (`microtime()` , `memory_get_peak_usage()`) to identify which parts of your code are consuming the most resources (time, memory). example code:

```
<?php
    $initialMemory = memory_get_peak_usage();

    //code or process here
    $largeArray = range(1, 1000000); // Creating a large array

    // Get memory usage after creating the large array
    $finalMemory = memory_get_peak_usage();

    // Calculate memory usage
    $memoryUsed = $finalMemory - $initialMemory;

    echo "Memory used: " . number_format($memoryUsed / 1024, 2) . " KB";
?>
```



- Optimize Database Query

Use appropriate indexes, optimize SQL queries, and minimize the number of queries executed. Avoid using `SELECT *` and fetch only required columns.

- Memory management

Unset variables or objects that are no longer needed to free up memory
(`unset($variable)`). example code:

```
<?php
    $myVariable = 'Hello, World!';
    echo "Before unset: $myVariable <br>";

    // Unset the variable
    unset($myVariable);
?>
```



- Optimize Loops and Iterations

- Use efficient loop constructs (`foreach` , `for` , `while`) and minimize unnecessary iterations.
- Break out of loops early if possible, using `break` or `continue` statements.