Voting Hub

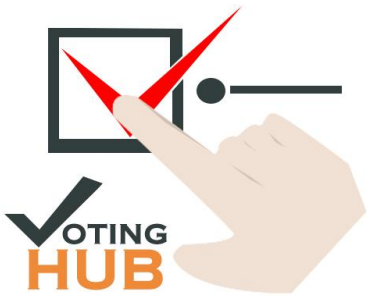
Table of Contents

1. App Name

- The application name “Voting Hub” conveys the idea of a platform where all voting activities can be held. The name “Voting Hub” also emphasizes that this application is specifically made for voting.

2. App Icon

- The Icon of the application was made to be simple and easily understood by the users of the application. Adding the name of the application just below the icon makes it more clear to the user that the application is made for voting.



- Checkbox with a check - This element represents the act of making a vote, similar to shading in a paper ballot. The check symbolizes the confirmation of the action of the voter.

- Index Finger - The index finger just on top of the checkbox represents the digital interaction between the application and the voter. It mimics the tapping on a screen emphasizing that the vote was made digitally instead of physically.

Circle with a line

- Mimicking a physical ballot number with the name of the candidate

Application Name Placement

- Positioned directly below the application to ensure the brand identification and to make it clear to the user that the application is a voting application.

Objectives

The objective of “Voting Hub” is to create a user friendly digital platform that even the elders can use to vote for their respectful candidate. Voting Hub also aim to be a versatile voting application where it can be a political voting or school voting or have a custom poll for specific event voting. Additionally, Voting Hub aims to be an accurate voting application where human errors would be reduced and provide a real time voting results.

Program Algorithm

1. User Authentication
    - User will need register an account including their voter’s ID to proceed to the Login section.

2. Display Dashboard
    - After register and logging in, the user will be transported to the dashboard where it contains a clickable sidebar to see the overall content of the application.

3. Display Polls
    - After clicking the polls, the application will display the ongoing and ended voting sessions.

4. Place Vote
    - The user then can proceed to choose their candidate and submit their vote real-time.

Program Listing

Register Web and Mobile

register.php

public_html > api > register.php

```php
<?php
include 'db.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {

    $firstName = trim($_POST['first_name']);
    $lastName = trim($_POST['last_name']);
    $birthday = trim($_POST['birthday']);
    $sex = trim($_POST['sex']);
    $cellphone = trim($_POST['cellphone']);
    $email = trim($_POST['email']);
    $voterId = trim($_POST['voter_id']);
    $password = trim($_POST['password']);


    if (empty($firstName) || empty($lastName) || empty($birthday) || empty($sex) || empty($cellphone) || empty($email) || empty($voterId) || empty($password)) {
        echo json_encode(["success" => false, "message" => "All fields are required."]);
        exit;
    }


    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        echo json_encode(["success" => false, "message" => "Invalid email format."]);
        exit;
    }


    if (strlen($password) < 8) {
        echo json_encode(["success" => false, "message" => "Password must be at least 8 characters long."]);
        exit;
    }


    if (!preg_match('/^\+?[0-9]{1,3}?[-.\s]?(\(?\d{1,4}?\)?[-.\s]?)*\d{4,}$/', $cellphone)) {
    echo json_encode(["success" => false, "message" => "Invalid phone number format."]);
    exit;
}


    $stmt = $conn->prepare("SELECT * FROM users WHERE email = ? OR voter_id = ?");
    $stmt->bind_param("ss", $email, $voterId);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        echo json_encode(["success" => false, "message" => "Email or Voter ID already exists."]);
        exit;
    }


    $hashedPassword = password_hash($password, PASSWORD_DEFAULT);


    $stmt = $conn->prepare("INSERT INTO users (first_name, last_name, birthday, sex, cellphone, email, voter_id, password) VALUES (?, ?, ?, ?, ?, ?, ?, ?)");
    $stmt->bind_param("ssssssss", $firstName, $lastName, $birthday, $sex, $cellphone, $email, $voterId, $hashedPassword);


    if ($stmt->execute()) {
        echo json_encode(["success" => true, "message" => "Registration successful"]);
    } else {
        echo json_encode(["success" => false, "message" => "Error: " . $stmt->error]);
    }


    $stmt->close();
} else {

    echo json_encode(["success" => false, "message" => "Invalid request method."]);
}

$conn->close();
?>
```

```kotlin
    private fun registerUser(
        firstName: String, lastName: String, birthday: String, sex: String,
        cellphone: String, email: String, voterId: String, password: String
    ) {
        val url = "https://votinghub.online/api/register.php"

        val client = OkHttpClient()
        val formBody = FormBody.Builder()
            .add( name: "first_name", firstName)
            .add( name: "last_name", lastName)
            .add( name: "birthday", birthday)
            .add( name: "sex", sex)
            .add( name: "cellphone", cellphone)
            .add( name: "email", email)
            .add( name: "voter_id", voterId)
            .add( name: "password", password)
            .build()

        val request = Request.Builder()
            .url(url)
            .post(formBody)
            .build()

        progressBar.visibility = ProgressBar.VISIBLE

        client.newCall(request).enqueue(object : Callback {
            override fun onFailure(call: Call, e: IOException) {
                runOnUiThread {
                    progressBar.visibility = ProgressBar.INVISIBLE
                    Toast.makeText( context: this@RegisterActivity, text: "Error: ${e.message}", Toast.LENGTH_SHORT).show()
                }
            }

            override fun onResponse(call: Call, response: Response) {
                runOnUiThread { progressBar.visibility = ProgressBar.INVISIBLE }
                if (response.isSuccessful) {
                    val responseBody = response.body?.string()
                    if (responseBody != null) {
```

Login Web and Mobile

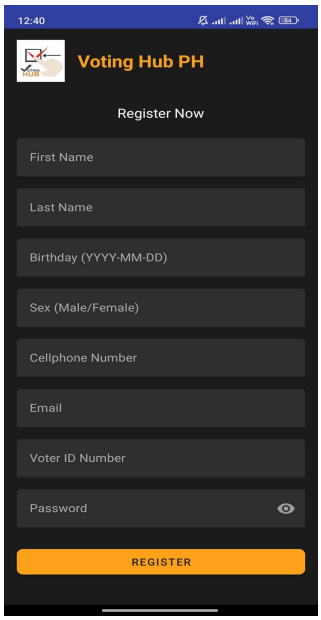
login.php

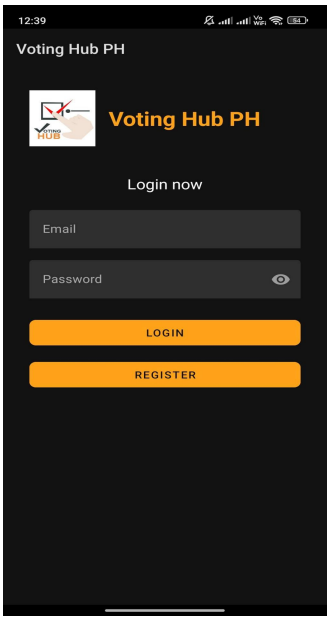
public_html > api > login.php

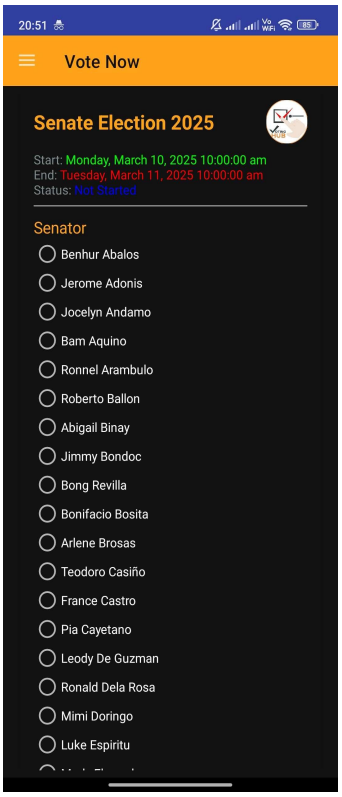
```php
<?php
session_start();
include 'db.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {

    $email = trim($_POST['email']);
    $password = trim($_POST['password']);

    if (empty($email) || empty($password)) {
        echo json_encode(["success" => false, "message" => "Email and password are required."]);
        exit;
    }

    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        echo json_encode(["success" => false, "message" => "Invalid email format."]);
        exit;
    }

    $stmt = $conn->prepare("SELECT id, password FROM users WHERE email = ?");
    if (!$stmt) {
        echo json_encode(["success" => false, "message" => "Database query error."]);
        exit;
    }

    $stmt->bind_param("s", $email);
    $stmt->execute();
    $stmt->store_result();

    if ($stmt->num_rows > 0) {
        $stmt->bind_result($user_id, $hashedPassword);
        $stmt->fetch();

        if (password_verify($password, $hashedPassword)) {

            $_SESSION['user_id'] = $user_id;
            echo json_encode(["success" => true, "message" => "Login successful"]);
        } else {
            echo json_encode(["success" => false, "message" => "Invalid password"]);
        }
    } else {
        echo json_encode(["success" => false, "message" => "User not found"]);
    }

    $stmt->close();
} else {
    echo json_encode(["success" => false, "message" => "Invalid request method."]);
}

$conn->close();
?>
```

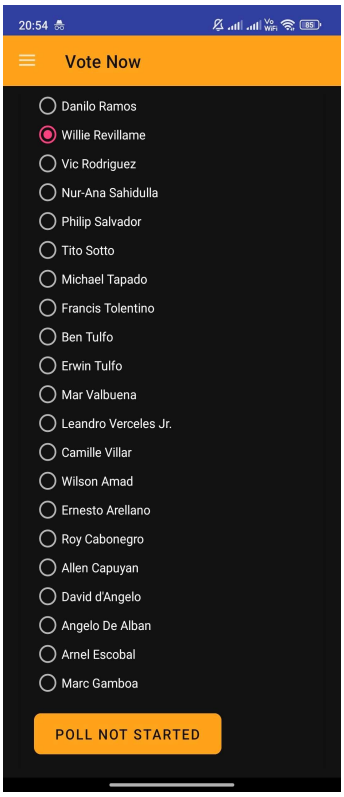
```
class MainActivity : AppCompatActivity() {
    private fun checkLoginStatus() {
    }

    private fun loginUser(email: String, password: String) {
        val url = "https://votinghub.online/api/login.php"
        val client = NetworkClient.client
        val formBody = FormBody.Builder()
            .add( name: "email", email)
            .add( name: "password", password)
            .build()

        val request = Request.Builder()
            .url(url)
            .post(formBody)
            .build()

        progressDialog.show()

        client.newCall(request).enqueue(object : Callback {
            override fun onFailure(call: Call, e: IOException) {
                runOnUiThread {
                    progressDialog.dismiss()
                    Toast.makeText( context: this@MainActivity,  text: "Network error: ${e.localizedMessage}", Toast.LENGTH_SHORT).show()
                }
            }

            override fun onResponse(call: Call, response: Response) {
                runOnUiThread { progressDialog.dismiss() }
                if (response.isSuccessful) {
                    val responseBody = response.body?.string()
                    if (responseBody == null) {
                        runOnUiThread {
                            Toast.makeText( context: this@MainActivity,  text: "Unexpected error occurred.", Toast.LENGTH_SHORT).show()
                        }
                        return
                    }
                    val json = JSONObject(responseBody)
                    val success = json.getBoolean( name: "success")
                    val message = json.getString( name: "message")
```

Program Output









Conclusion

In conclusion, We created “Voting Hub” to lessen the manual work of voting by implementing a digitalize voting platform. Digital voting platform helps minimize the reliance on physical ballots and manual counting resulting in reducing of physical labor while maintaining the accuracy of the voting process.

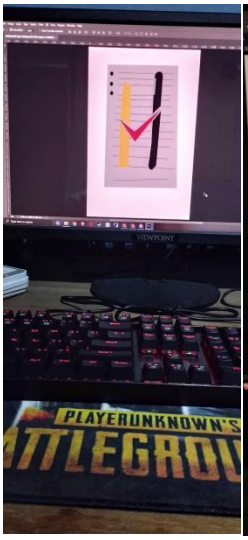
Gantt Chart Documentation

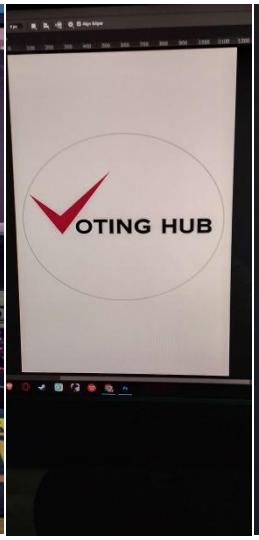
November 26, 2024 - December 4, 2024

We had to think carefully what application we can handle that will be easier for the both of us and it resulted to a voting application which ended up being our project of choice.

December 6, 2024 - December 7, 2024

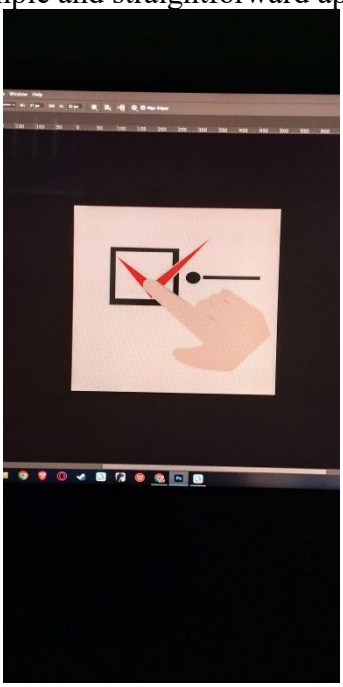
Created some logo to feature the mobile application at night. Did not like the outcome, be it complex or simple design, so we have to make some options.

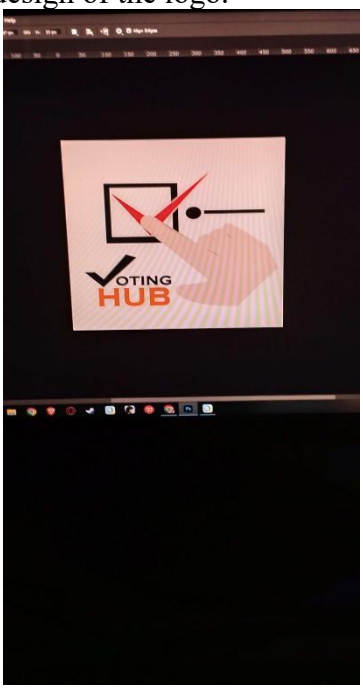




December 9, 2024

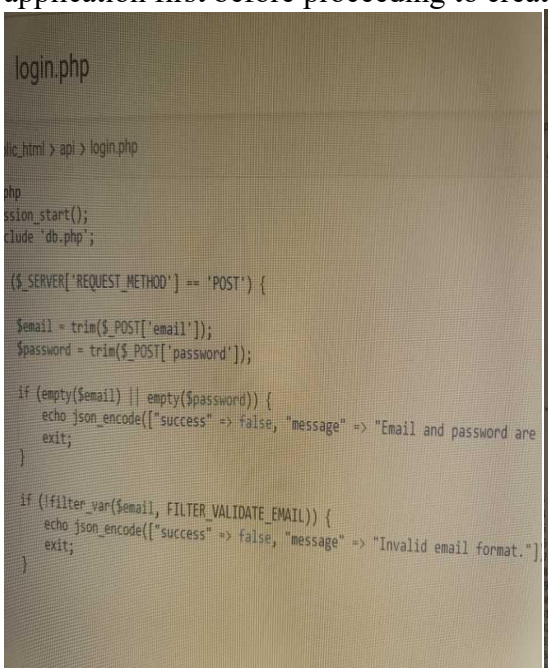
After some works, we decided finalize the application logo by going with a more simple and straightforward approach to the design of the logo.

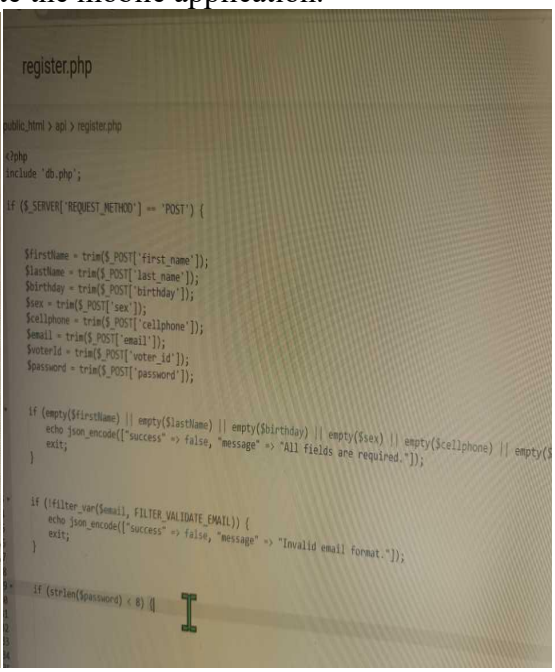




December 10, 2024 - December 11, 2024

For this 2 days, we did the back-end of the web version of the voting application first before proceeding to create the mobile application.



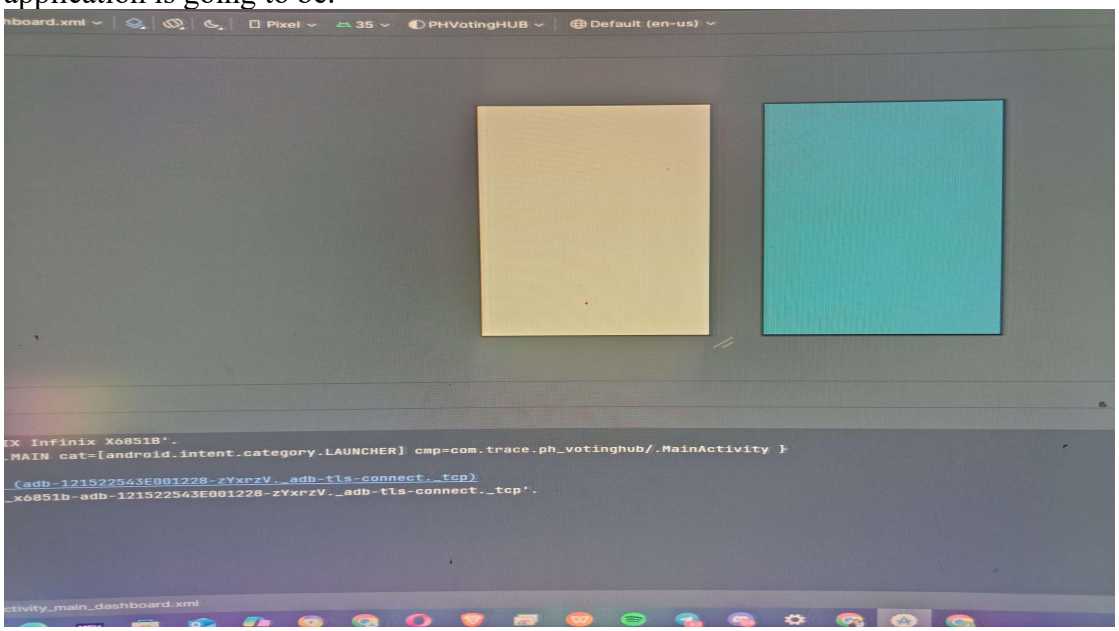


December 13, 2024 - December 15, 2024

We created the admin panel and the needed database in order for us to connect the web version to the mobile application.

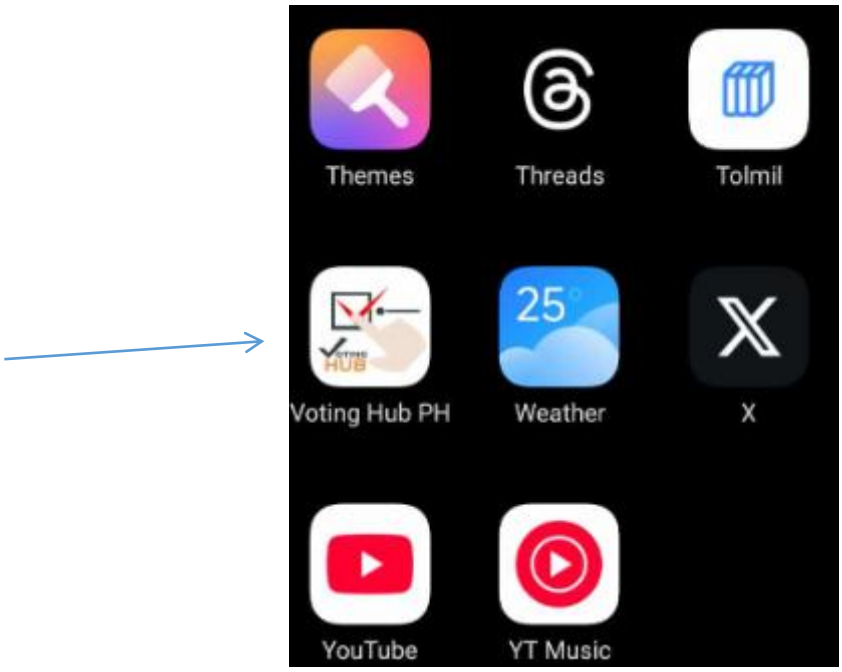
December 17, 2024 - December 19, 2024

Explored Android Studio to know where to start and how the flow of the application is going to be.



January 6, 2025 - January 7, 2025

By exploring android studio, we managed to create a simple mobile application that we can connect the API

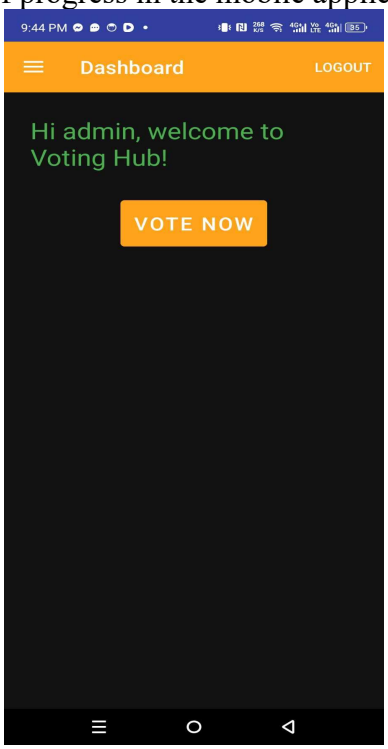


January 8, 2025 - January 9, 2025

Since the web and mobile is created, we managed to connect the api of both web and mobile to share the database recorded.

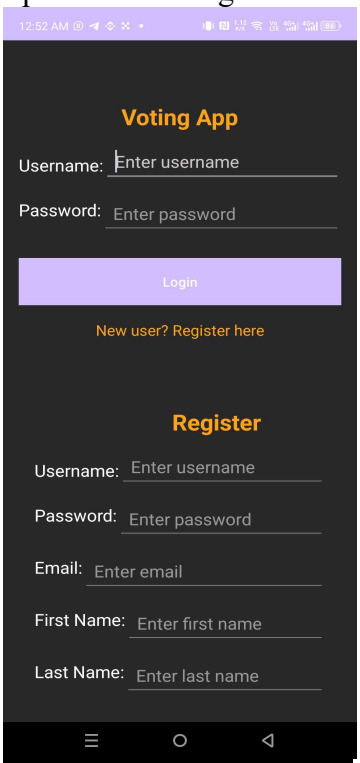
January 17, 2025

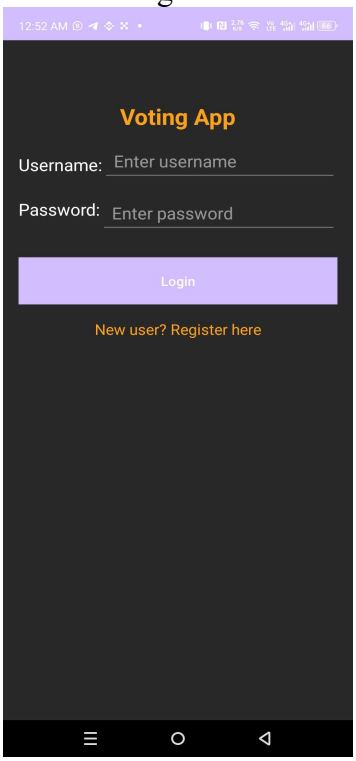
Examination came and we just managed to create a simple dashboard to proceed with the checking of progress in the mobile application project.

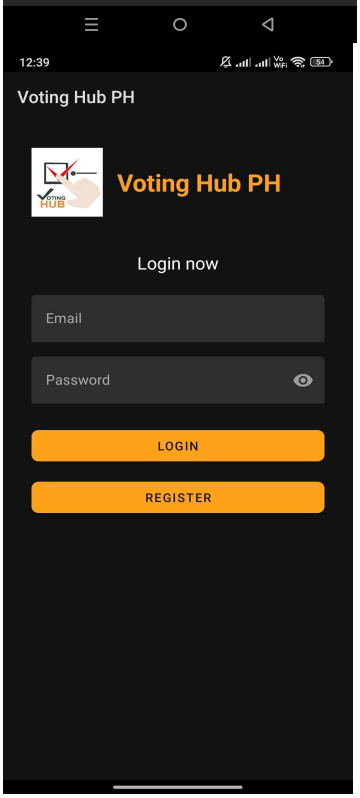


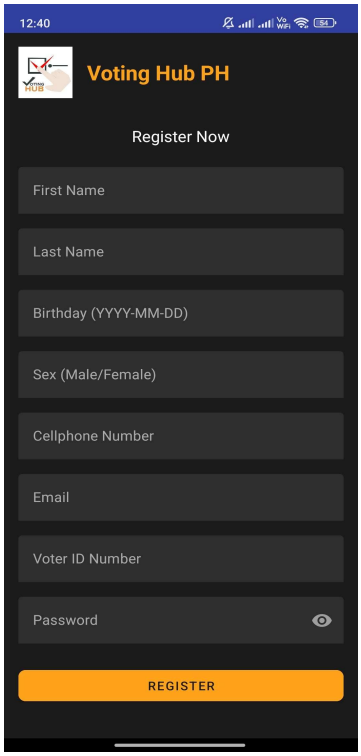
January 25, 2025 - January 26, 2025

Improved the design of the mobile UI to make it organized



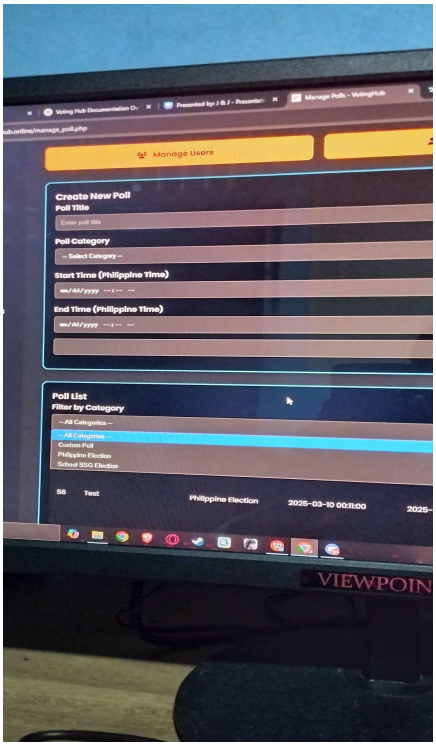






February 15, 2025 - February 18, 2025

Started adding some voting contents such as school student council and political elections

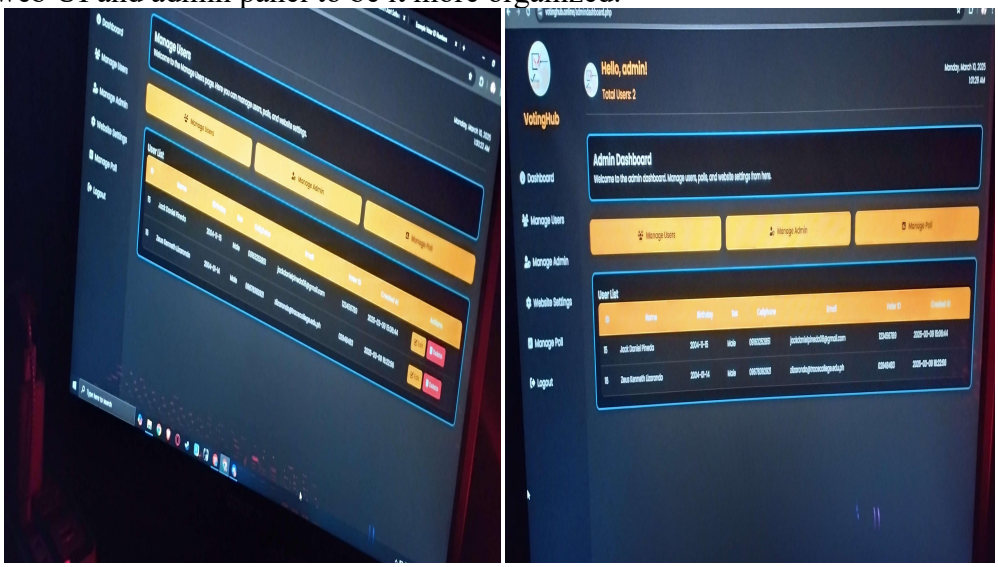


February 22, 2025 - February 26, 2025

We created the whole back-end of the mobile application to finish the whole function of each element so that we would just need to focus on the design of the application itself.

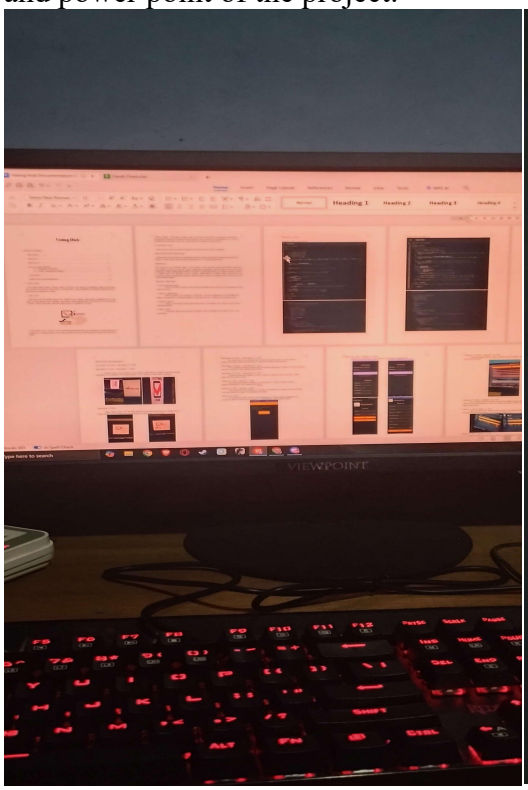
March 1, 2025 - March 2, 2025

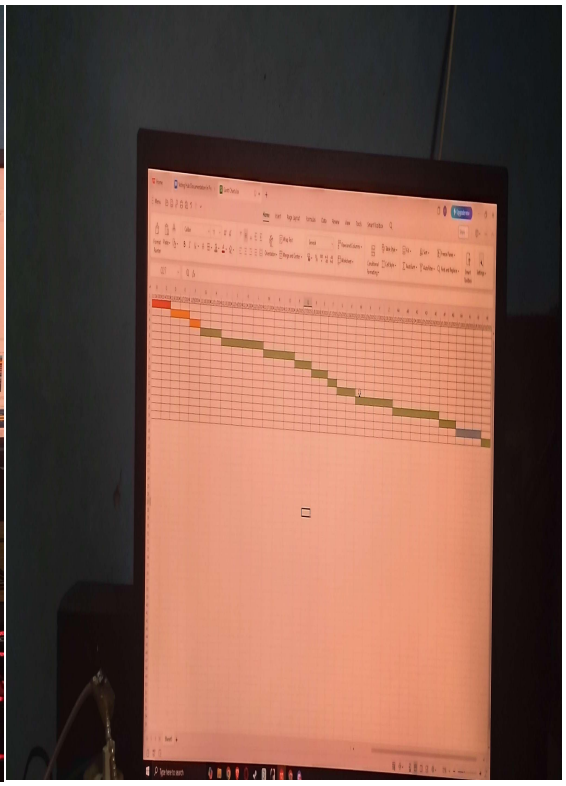
Upon finishing the design of the mobile, We further improved the design of the web UI and admin panel to be it more organized.



March 7, 2025 - March 9, 2025

Upon completing the application, we proceeded to create the documentation and power point of the project.





March 10, 2025

Added a multiple voting since the current user can only vote 1 per user.