

T.W.I.P.

Two-Wheeler Indicator Panel

Feedback

Add a way to retrieve your account if you lose your pincode.

Emphasize what makes our product unique.

Delve deeper into details about how we are going to realize everything.



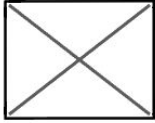
Organization & Planning

Front-end

Back-end



Product increment - Log In page



Username

Pincode

No account? Sign up!

Two Wheeler Indicator Panel

Log In

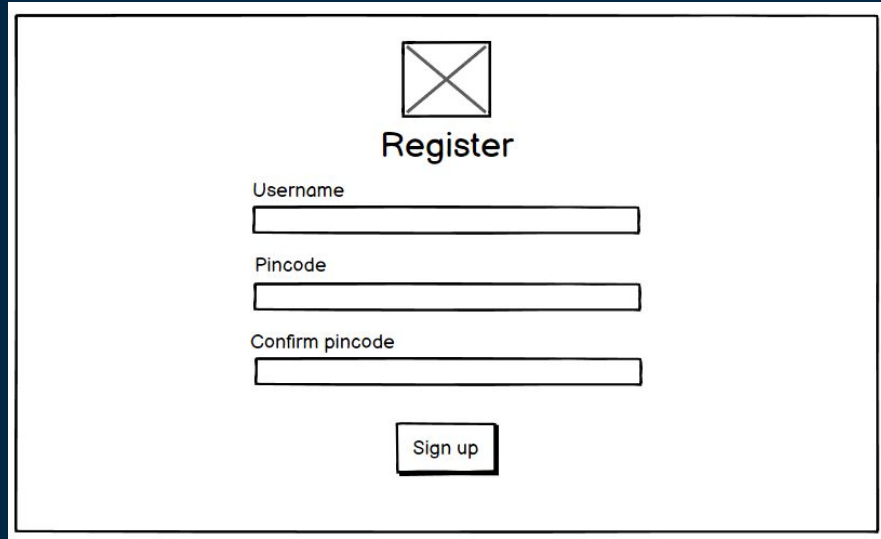
Don't have an account? [Sign Up](#)

Username:


Pincode:

[Forgot your Pincode?](#)

Product increment - Register page



A wireframe diagram of a registration page. At the top center is a square icon with an 'X' inside. Below it is the title 'Register'. The form consists of three input fields: 'Username', 'Pincode', and 'Confirm pincode'. Each field is represented by a horizontal rectangle. Below the input fields is a 'Sign up' button, represented by a rounded rectangle.

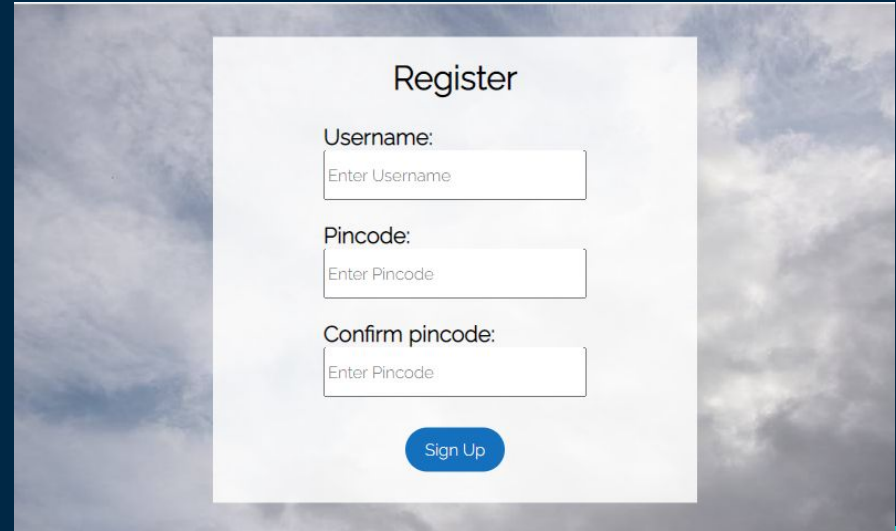


Register

Username

Pincode

Confirm pincode



A mockup of a registration page. The background is a cloudy sky. The registration form is a white card with the title 'Register'. It contains three input fields: 'Username' with placeholder text 'Enter Username', 'Pincode' with placeholder text 'Enter Pincode', and 'Confirm pincode' with placeholder text 'Enter Pincode'. Below the input fields is a blue 'Sign Up' button.

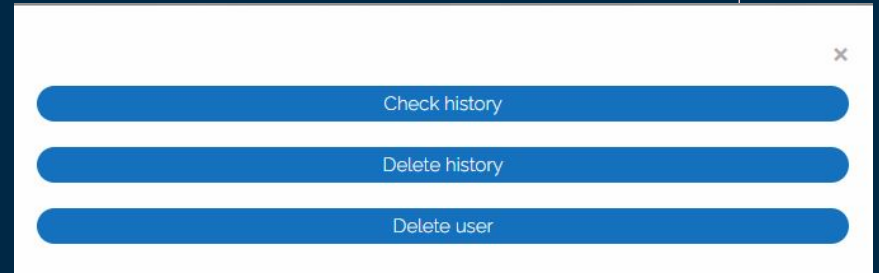
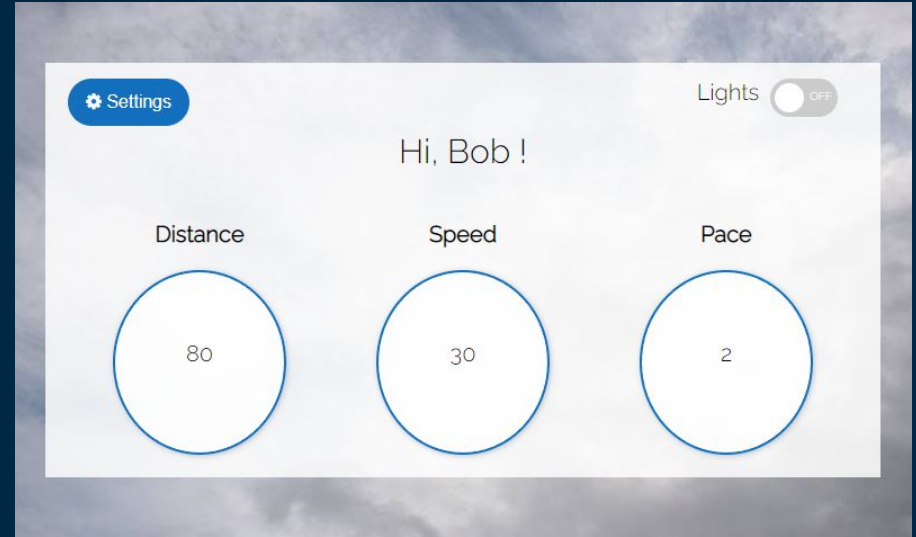
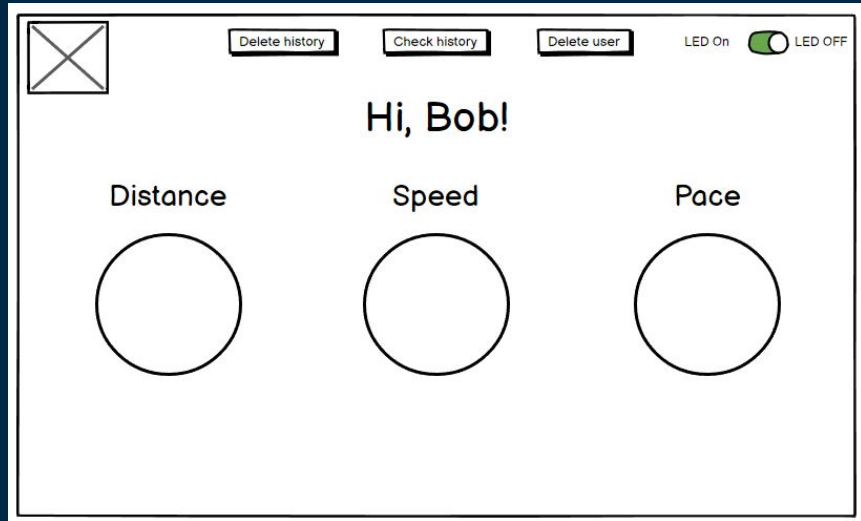
Register

Username:

Pincode:

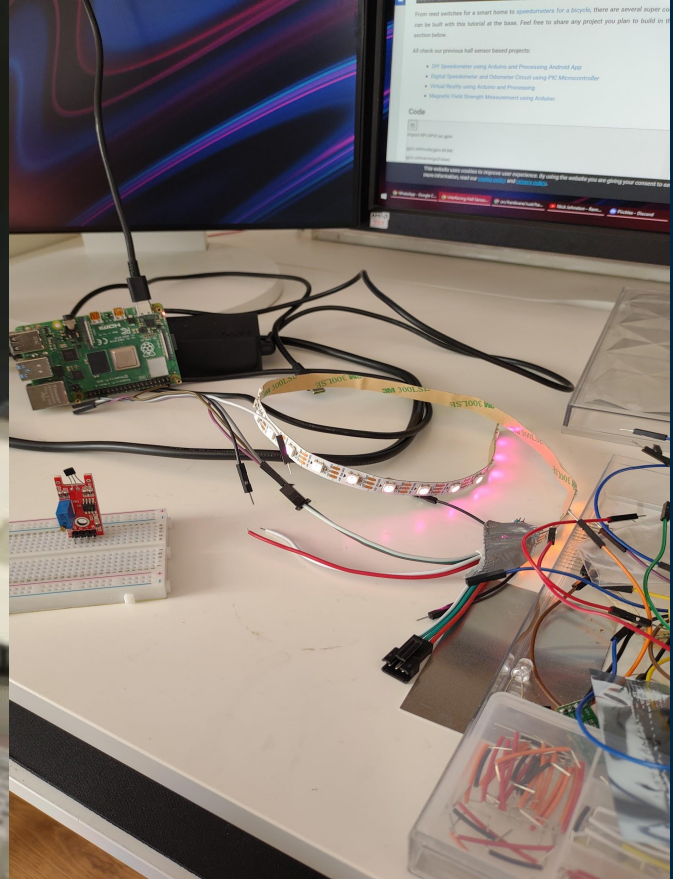
Confirm pincode:

Product increment - Homepage

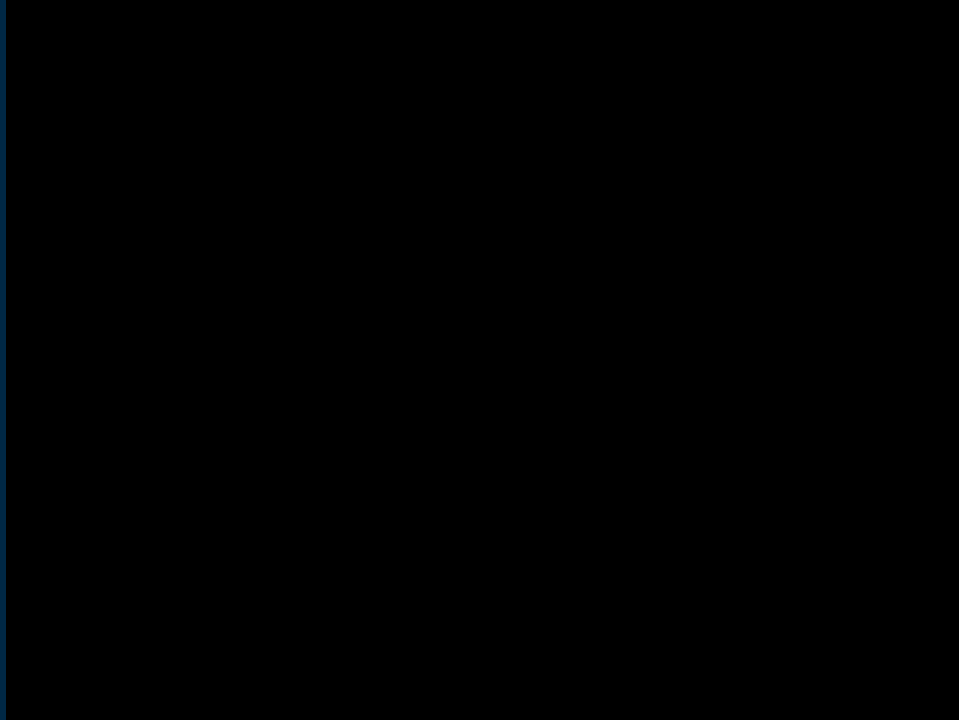


Sensor

```
readhall.py
1 import RPi.GPIO as gpio
2 import time
3
4 gpio.setmode(gpio.BCM)
5 gpio.setup(0, gpio.IN, pull_u
6
7 while True:
8     print(gpio.input(0))
9     time.sleep(1)
```



MVP



Programming

Backend

Hardware



Testing

Unit Testing (rust)

System Testing

Hardware Testing

```
#[cfg(test)]
mod tests {
    use super::rocket;
    use super::HardwareData;
    use rocket::http::Status;
    use rocket::local::Client;

    // Testing whether the endpoint contains "distance", "speed" and "pace" and wheter
    #[test]
    Run test
    fn test_generate_hardware_data() {
        let client = Client::new(rocket()).expect("valid rocket instance");
        let mut response = client.get("/hardware/data").dispatch();

        assert_eq!(response.status(), Status::Ok);
        let message = response.body_string().unwrap();
        assert_eq!(message.contains("distance"), true);
        assert_eq!(message.contains("speed"), true);
        assert_eq!(message.contains("pace"), true)
    }

    // Test which sees if the HardwareData got created correctly
    #[test]
```

Security

Authentication

System security

Security tests



Planning for next sprint

Improve Front-end

Making a logo

User authentication

Show trip history

Light sensor



Q&A

The background is a dark blue gradient. It features an abstract pattern of small squares and thin vertical lines in various colors including teal, orange, pink, and light blue. These elements are scattered across the slide, with some appearing as solid shapes and others as outlines. The central text 'Q&A' is in a white, serif font.