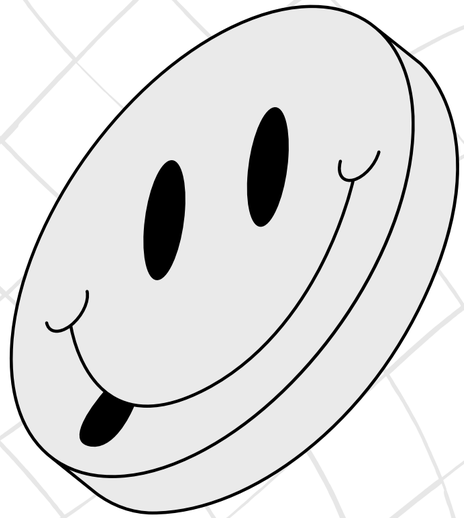
ONLINE LEARNING



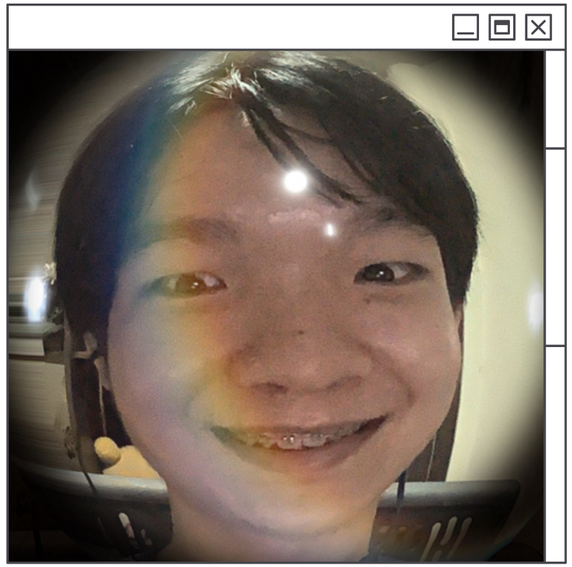
Mobile & Cloud Computing

MEMBER

Programmer, Tester

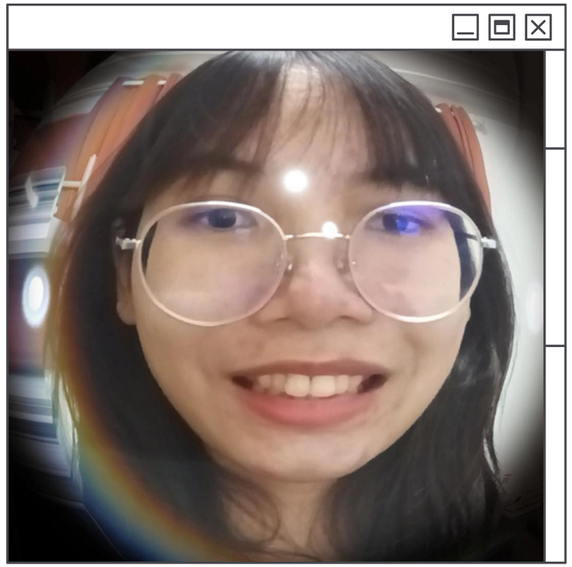
6303051623209
นายวัชระ ศรีวิชัย

Project Manager, Tester



6303051613106
นายจิรภัทร ศรีสมพันธุ์

Front-end, Presentation



6303051623179
นางสาวศิริวรรณ ทุหา

System Analysis

6303051623217
นายจักรภัทร คำนารักษ์

```rust
#[get("/manage/user")]
pub async fn view_user_score() -> impl Responder {
    let response_body: Value = json!(
        {
        "massage": "List of the user",
        "user": [
          {
            "us_name": "Loki",
            "us_score": "50"
          },
          {
            "us_name": "Lola",
            "us_score": "60"
          },
          {
            "us_name": "Lucy",
            "us_score": "79"
          }
        ]
      }
    );

    HttpResponse::Ok().json(response_body)
}
```

Admin handler

- View user score

Status code : 200

Admin handler

- Delete user

Status code : 204

```rust
#[delete("/manage/user")]
pub async fn delete_user() -> impl Responder {
    HttpResponse::NoContent().finish()
}
```

Admin handler

- Upload course

Status code : 201

```rust
#[post("/manage/course/{id}")]
pub async fn upload_course() -> impl Responder {
    HttpResponse::Created().json("Upload complete")
}
```

Admin handler

- Delete course

Status code : 204

```rust
#[delete("/manage/course/{id}")]
pub async fn delete_course() -> impl Responder {
    HttpResponse::NoContent().finish()
}       You, 3 days ago • apiary_complete
```

User handler

- Login

Status code : 200

Status code : 401

```rust
#[post("/login")]
pub async fn login(user: web::Json<User>) -> impl Responder {
    // perform authentication
    if user.username == "username" && user.password == "password" {
        HttpResponse::Ok().finish()
    } else {
        HttpResponse::Unauthorized().finish()
    }
}
```

User handler

- **Signup**

Status code : 201

```rust
#[post("/signup")]
pub async fn signup(user: web::Json<User>) -> impl Responder {
    // Implement user registration logic here, for example, by adding the user to a database
    let response: String = format!("User with username {} successfully registered!",user.username);
    let response_body: Value = json!(response);
    HttpResponse::Created().json(response_body)
}
```

User handler

- View a course

Status code : 200

Status code : 404

```rust
#[get("/courses/{id}")]
async fn get_course(course_id: web::Path<i32>) -> impl Responder {
    let database: Database = Database::new(); // create a new instance of Database
    let course: Option<&Course> = database.get_course(course_id.into_inner()); // pass it

    match course {
        Some(course: &Course) => HttpResponse::Ok().json(course),
        None => HttpResponse::NotFound().finish(),
    }
}       You, 5 days ago • learning_V.1
```

User handler

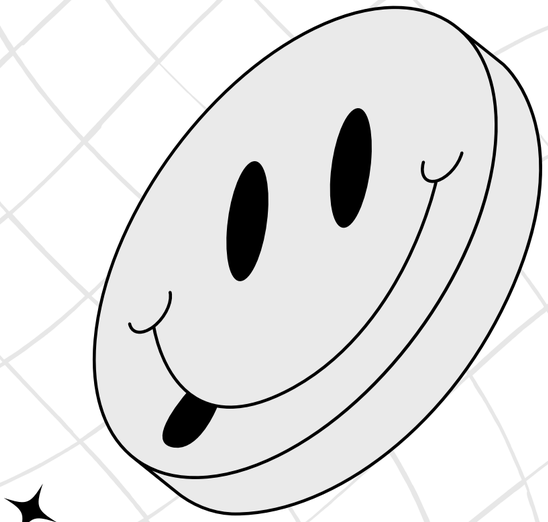
- View all course

Status code : 204

```rust
#[get("/courses")]
async fn get_all_courses() -> impl Responder {
    let database: Database = Database::new(); // create a new instance of Database
    let courses: &Vec<Course> = database.get_courses(); // pass it to the function

    HttpResponse::Ok().json(courses)
}
```

Thank



You