**Working Report – Phan Huu Hung**

**1. Project Setup**

1.1. Environment:

- OS: MacOS Catalina

- Go: 1.13.4

- IDE: GoLand

- DB: DynamoDB Local run on Docker with sharedDb configuration

1.2. Project

- Main moduldes:

* aws-sdk-go/service/dynamodb: For connect to DynamoDB
* gorilla/mux: For Go Webserver, using it because it’s light and simple when comparing with revel
* encoding/json, net/http, encoding/base64: For handle request, response in json
* joho/godotenv: For reading .env configuration file
* crypto/sha256, google/uuid: For encrypted password, and generating token

- Created Table Script: **dynamodb\_script.md in docs directory**

- Configuration: Using **.env** file that provide connection to DynamoDB Local

- Project structure: using MVC model with Controllers (User), Object Models DTO (User, Response), DAO (DynamoDB Client Connection and Operation)

- Source Code, Project Management: Github (<https://github.com/hungph/go-webserver>), Task Management – Github (<https://github.com/hungph/go-webserver/projects/1>)

- Code structure:

|  |
| --- |
| A screenshot of a cell phone  Description automatically generated |

**2. Code Weakness**

- This is my very first time to work with Go, I don’t know Go Lang and it’s eco-system before, so my code may have many, many bugs even I tried my best.

**3. Next Tasks**

- This is very simple sign-up, sign-in system. For using this in production, we can move to use OAuth2 for authentication and authorization user in our system.