# Flask REST API Code

Coded by Muhammad Essa

## 1. Importing Libraries

The necessary libraries like Flask, flask\_restful, flask\_jwt\_extended, SQLAlchemy, and werkzeug.security are imported. These libraries are used to build the API, handle requests, manage security with JWT, and connect to the database.

## 2. Flask App Initialization

The Flask application is initialized, and flask\_restful is used to create API resources. The JWT is configured for authentication, and SQLAlchemy is set up to interact with an SQLite database.

## 3. Database Models

Two database models are defined:   
- User: Contains id, username, and password fields.   
- Item: Contains id, name, and an optional description field.

## 4. Database Creation

The database tables are created using SQLAlchemy, and the app's context ensures the tables are generated on startup.

## 5. User Registration

A Register resource class is defined to handle user registration. It checks if a user already exists, hashes the password using generate\_password\_hash, and adds the user to the database.

## 6. User Login

A Login resource class is defined for user authentication. It verifies the user's credentials and, if correct, returns a JWT access token, allowing the user to access protected resources.

## 7. CRUD Operations for Items

The ItemResource class allows the user to perform Create, Read, Update, and Delete (CRUD) operations on items. All operations require JWT authentication to ensure only authorized users can modify the data.

## 8. Routes

Routes are added to map URLs to the resource classes:  
- /register: For user registration.  
- /login: For user login.  
- /item/<int:item\_id>: To manage CRUD operations for items.