Student Task: Implement Flask API with Login, 2FA, and JWT Authentication

Task Overview:

Build a secure Flask API connected to a MySQL to handle user registration, login with Two-Factor Authentication (2FA) using Google Authenticator, and issue JWT tokens to authenticate CRUD operations on a separate database table (`Products`).

Task Steps:

1. User Authentication:

- Create a Flask API endpoint for **user registration**:
  - User provides a username and password.
  - Hash the password securely and store it in MySQL along with a secret key for Google Authenticator (2FA).

2. Setup Google Authenticator for 2FA:

- Generate a **QR Code** via Flask endpoint for the user to scan using Google Authenticator.
- Implement an API endpoint to verify the code provided by the user's Google Authenticator app.

3. Login and JWT Token Generation:

- User logs in by providing username and password.
- Verify credentials; if correct, prompt user for 2FA code from Google Authenticator.
- After successful verification, generate a JWT token valid for **10 minutes**.

4. JWT-Secured CRUD Operations:

- Implement secured CRUD operations (Create, Read, Update, Delete) using JWT for authentication on a `Products` table.

Database Structure:

Users Table

Column	Data Type	Description
id	INT (Identity)	Primary Key
username	VARCHAR(50)	Unique username
password	VARCHAR(256)	Hashed password

Column	Data Type	Description
twofa_secret	VARCHAR(256)	Google Authenticator secret key

Products Table for CRUD Operations:

Column	Data Type	Description
id	INT (Primary)	Auto-increment product ID
name	VARCHAR(100)	Product name
description	VARCHAR(255)	Product description
price	DECIMAL(10,2)	Price
quantity	INT	Stock quantity

- Implement secured endpoints to:
  - **Create** new products
  - **Read** existing products
  - **Update** product information
  - **Delete** products