**Name: Astha Nitinkumar Patel**

**University name: Manhattan University**

**Subject name: Software Engineering  
30 Nov 2024**

1. Secure login functionality for students.
2. Faculty members can upload/manage course materials, assignments, and grades.

### ****Directory Structure****

arduino

project/

│

├── app.py

├── templates/

│ ├── login.html

│ ├── dashboard.html

│ ├── upload.html

├── static/

│ ├── style.css

├── requirements.txt

├── database.db

### ****Code Implementation****

#### app.py

python

from flask import Flask, render\_template, request, redirect, url\_for, session, flash

from werkzeug.security import generate\_password\_hash, check\_password\_hash

import sqlite3

app = Flask(\_\_name\_\_)

app.secret\_key = 'your\_secret\_key'

# Database initialization

def init\_db():

with sqlite3.connect('database.db') as conn:

cursor = conn.cursor()

cursor.execute('''

CREATE TABLE IF NOT EXISTS users (

id INTEGER PRIMARY KEY AUTOINCREMENT,

username TEXT NOT NULL UNIQUE,

password TEXT NOT NULL,

role TEXT NOT NULL

)

''')

cursor.execute('''

CREATE TABLE IF NOT EXISTS course\_materials (

id INTEGER PRIMARY KEY AUTOINCREMENT,

title TEXT NOT NULL,

description TEXT,

uploaded\_by TEXT NOT NULL

)

''')

print("Database Initialized")

init\_db()

# Route: Home

@app.route('/')

def home():

return render\_template('login.html')

# Route: Login

@app.route('/login', methods=['GET', 'POST'])

def login():

if request.method == 'POST':

username = request.form['username']

password = request.form['password']

with sqlite3.connect('database.db') as conn:

cursor = conn.cursor()

cursor.execute('SELECT \* FROM users WHERE username = ?', (username,))

user = cursor.fetchone()

if user and check\_password\_hash(user[2], password):

session['username'] = user[1]

session['role'] = user[3]

if user[3] == 'faculty':

return redirect(url\_for('faculty\_dashboard'))

return redirect(url\_for('student\_dashboard'))

else:

flash("Invalid credentials!")

return render\_template('login.html')

# Route: Student Dashboard

@app.route('/student/dashboard')

def student\_dashboard():

if 'username' in session and session['role'] == 'student':

return render\_template('dashboard.html', username=session['username'])

return redirect