

Uitleg code

Login scherm

```
<body>
  <div class="login-container">
    <h2>Admin Login</h2>
    <form class="login-form" action="/login" method="POST">
      <input type="text" name="username" placeholder="Username" required>
      <input type="password" name="password" placeholder="Password" required>
      <button type="submit">Login</button>
    </form>
    <p>Heb je nog geen account? Vraag een admin om een account voor je te maken!</p>
  </div>
</body>
</html>
```

Met deze code heb ik een form gemaakt waar mensen hun username en wachtwoord kunnen invullen wanneer ze dat hebben gedaan kunnen ze op de knop login klikken en gaan ze naar de home pagina

Register scherm

```
<body>
  <div class="register-container">
    <h2>Register Admin</h2>
    <form class="register-form" method="POST">
      <input type="text" name="username" placeholder="Username" required>
      <input type="password" name="password" placeholder="Password" required>
      <button type="submit">Register</button>
    </form>
  </div>
</body>
</html>
```

Bij deze code heb ik een form gemaakt waar mensen zich kunnen registreren. Bij de eerste input vraag ik voor een username en bij de tweede input vraag ik voor een password. De knop register brengt je terug naar het login scherm.

Aanmaken van de database

```
def init_admin_db():
    conn = sqlite3.connect('admin_database.db')
    cursor = conn.cursor()

    cursor.execute('''
        CREATE TABLE IF NOT EXISTS admins (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            username TEXT UNIQUE NOT NULL,
            password TEXT NOT NULL
        )
    ''')

    conn.commit()
    conn.close()
```

Met deze code wordt de database aangemaakt

≡ admin_database.db

Dit is de aangemaakte database

Register password hash

```
def register_admin(username, password):
    hashed_password = generate_password_hash(password)
    conn = sqlite3.connect('admin_database.db')
    cursor = conn.cursor()

    try:
        cursor.execute('INSERT INTO admins (username, password) VALUES (?, ?)', (username, hashed_password))
        conn.commit()
        return True
    except sqlite3.IntegrityError:
        return False
    finally:
        conn.close()
```

Met deze code zet ik de connectie aan de database en zorg ik ervoor dat de passwords gehashed worden

Gegevens ophalen

```
def get_admin(username):  
    conn = sqlite3.connect('admin_database.db')  
    cursor = conn.cursor()  
    cursor.execute('SELECT * FROM admins WHERE username = ?', (username,))  
    admin = cursor.fetchone()  
    conn.close()  
    return admin
```

Met deze code haal ik de username gegevens op

Check of gegevens kloppen

```
def verify_admin(username, password):  
    admin = get_admin(username)  
    print(f"Admin found: {admin}")  
    if admin and check_password_hash(admin[2], password):  
        print("Password matches")  
        return True  
    else:  
        print("Password does not match")  
    return False
```

Met deze code kijk ik of de username en password kloppen en of de user doorgestuurd mag worden

Check of admin klopt

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']

        if verify_admin(username, password):
            session['admin'] = True
            flash('Welcome, Admin!', 'success')
            return redirect(url_for('index'))
        else:
            flash('Invalid credentials', 'danger')

    return render_template('login.html')
```

Deze code checked of er ingelogd is als de gegevens kloppen wordt je doorgestuurd naar de home pagina als ze niet kloppen wordt je terug gestuurd naar de inlog pagina

Check of je admin bent

```
def admin_required(f):
    @wraps(f)
    def decorated_function(*args, **kwargs):
        if not session.get('admin'):
            flash('You need to login first!', 'warning')
            return redirect(url_for('login'))
        return f(*args, **kwargs)
    return decorated_function
```

Dit checked en zorgt ervoor dat alleen admins bepaalde routes kunnen volgen

Toegang voor admin admin

`@admin_required`

Dit zorgt ervoor dat je alleen deze route kan volgen als admin ben je dit niet wordt je terug gestuurd naar de login