```
from flask import Flask, flash, request, session
from flask.templating import render_template
from werkzeug.utils import redirect, secure_filename
import bcrypt
from functions.users import check_existing_user, add_new_user, check_user_credentials
from functions.scheduler import add_medicine, fetch_user_schedule, card, no_data_check
from functions.dashboard import user_name, today_card, tomorrow_card, day_after_card
from functions.medicines import get_all_medicines, medicine_card
from functions.medscraper import get_medicines
from functions.prescription import add_prescription, add_prescription_record, fetch_prescriptions,
prescription_card
from functions.notifications import *
from datetime import datetime, timedelta
import os

app = Flask(__name__)
app.secret_key = 'subhogay'
app.config['UPLOAD_FOLDER'] = 'uploads'
app.config['MAX_CONTENT_LENGTH'] = 20 * 1000 * 1000



@app.route('/')
def index():
    try:
        if session['login']:
            login = True
        else:
            login = False
    except KeyError:
        session['login'] = False
        login = session['login']
```

```
    return render_template('index.html', login=login)



# Authentication
@app.route('/signup')
def signup():
    return render_template('signup.html')



@app.route('/signup/verify', methods=['GET', 'POST'])
def signup_verify():
    if request.method == 'POST':
        data = request.form
        name = data['name']
        email = data['email']
        contact = data['contact']
        emergency_contact = data['emergency_contact']
        password = data['password'].encode()
        hashed = bcrypt.hashpw(password, bcrypt.gensalt())

        if not check_existing_user(email):
            add_new_user(name, email, hashed, contact, emergency_contact)
            flash('Registered')
            return redirect('/')
        else:
            flash('User with this email already exists.', 'error')
            return render_template('signup.html')



@app.route('/login')
def login():
```

```
    try:
        if session['login']:
            return redirect('/dashboard')
        else:
            return render_template('login.html')
    except:
        return render_template('login.html')


@app.route('/login/verify', methods=["GET", "POST"])
def login_verify():
    if request.method == "POST":
        data = request.form
        email = data['email']
        password = data['password']
        if check_user_credentials(email, password):
            session['user'] = email
            session['login'] = True
            flash('Logged in')
            return redirect('/dashboard')
        else:
            flash('Incorrect username or password!', 'error')
            return render_template('login.html')


@app.route('/logout')
def logout():
    session.clear()
    session['login'] = False
    session['user'] = None
    return redirect('/login')
```

```
@app.route('/dashboard')
def dashboard():
    try:
        if session['user']:
            if not no_data_check(session['user']):
                data = fetch_user_schedule(datetime.now(), session['user'])
                now = str(datetime.now()).split('.')[0][-8:-3]
                data['current_time'] = now

                upcoming = {}
                completed = {}

                data = dict(sorted(data.items(), key=lambda item: item[1]))

                upcoming_count = 0
                completed_count = 0
                checkpoint = False
                for medicine in data:
                    if medicine == 'current_time':
                        checkpoint = True
                    elif checkpoint:
                        upcoming_count += 1
                        upcoming[medicine] = data[medicine]
                    else:
                        completed_count += 1
                        completed[medicine] = data[medicine]

                today_card_html = ''
                for medicine in upcoming:
```

```
            today_card_html += today_card(medicine, upcoming[medicine])


        first = 0
        for medicine in upcoming:
            first += 1
            if first < 2:
                first_medicine = medicine
                first_medicine_time = upcoming[medicine]


            card1 = f"""<div class="md:p-7 p-4">
                    <h2 class=" text-xl text-center text-primary-green-dark capitalize">Next Dose</h2>
                    <h3 class="text-sm  text-primary-green-dark  text-center">{first_medicine_time} -
{first_medicine}</h3>
                </div>"""


            card2 = f"""<div class="md:p-7 p-4">
                    <h2 class="text-xl text-center text-primary-blue-dark capitalize">Today</h2>
                    <h3 class="text-sm  text-primary-blue-dark  text-center">{upcoming_count +
completed_count} doses</h3>
                </div>"""


            card3 = f"""<div class="md:p-7 p-4">
                    <h2 class="text-lg text-center text-primary-yellow-dark capitalize">
                      <span>{first_medicine}</span>
                    </h2>
                    <h3 class="text-sm  text-primary-yellow-dark  text-
center">{first_medicine_time}</h3>
                </div>"""


        count = 0
        tomorrow_card_html = ''
        data = fetch_user_schedule(datetime.now() + timedelta(days=1), session['user'])
```

```
        data = dict(sorted(data.items(), key=lambda item: item[1]))
        for medicine in data:
            count += 1
            if count < 4:
                tomorrow_card_html += tomorrow_card(medicine, data[medicine])


        count = 0
        day_after_card_html = ''
        data = fetch_user_schedule(datetime.now() + timedelta(days=2), session['user'])
        data = dict(sorted(data.items(), key=lambda item: item[1]))
        for medicine in data:
            count += 1
            if count < 4:
                day_after_card_html += day_after_card(medicine, data[medicine])
        session['upcoming_count'] = upcoming_count


        if not upcoming and not completed:
            card1 = f"""<div class="md:p-7 p-4">
                            <h2 class=" text-xl text-center text-primary-green-dark capitalize">Add some
data from scheduler</h2>
                        </div>"""
            card2 = f"""<div class="md:p-7 p-4">
                             <h2 class="text-xl text-center text-primary-blue-dark
capitalize">Today</h2>
                             <h3 class="text-sm  text-primary-blue-dark  text-center">{upcoming_count
+ completed_count} doses</h3>
                            </div>"""
            return render_template('app/dashboard.html', name=user_name(session['user']).title(),
card1=card1,
                                   card2=card2, upcoming_count=upcoming_count,
today_card_html=today_card_html,
                                   tomorrow_card_html=tomorrow_card_html,
                                   day_after_card_html=day_after_card_html)
```

```
            elif not upcoming:
                card2 = f"""<div class="md:p-7 p-4">
                                    <h2 class="text-xl text-center text-primary-blue-dark
capitalize">Today</h2>
                                    <h3 class="text-sm  text-primary-blue-dark  text-center">{upcoming_count
+ completed_count} doses</h3>
                                </div>"""
                return render_template('app/dashboard.html', name=user_name(session['user']).title(),
                                    card2=card2, upcoming_count=upcoming_count,
                                    today_card_html=today_card_html,
                                    tomorrow_card_html=tomorrow_card_html,
                                    day_after_card_html=day_after_card_html)
            else:
                return render_template('app/dashboard.html', name=user_name(session['user']).title(),
card1=card1,
                                    card2=card2,
                                    card3=card3, upcoming_count=upcoming_count,
today_card_html=today_card_html,
                                    tomorrow_card_html=tomorrow_card_html,
                                    day_after_card_html=day_after_card_html)
        else:
            card1 = f"""<div class="md:p-7 p-4">
                                <h2 class=" text-xl text-center text-primary-green-dark capitalize">Add some
data from scheduler</h2>
                            </div>"""

            return render_template('app/dashboard.html', name=user_name(session['user']).title(),
card1=card1,
                                upcoming_count=0)
    else:
        return redirect('/login')
  except KeyError:
    return redirect('/login')
```

```
@app.route('/schedule')
def schedule():
    try:
        if session['user']:
            data = fetch_user_schedule(datetime.now(), session['user'])
            now = str(datetime.now()).split('.')[0][-8:-3]
            data['current_time'] = now

            upcoming = {}
            completed = {}

            data = dict(sorted(data.items(), key=lambda item: item[1]))

            checkpoint = False
            for medicine in data:
                if medicine == 'current_time':
                    checkpoint = True
                elif checkpoint:
                    upcoming[medicine] = data[medicine]
                else:
                    completed[medicine] = data[medicine]

            upcoming_count = 0
            upcoming_card_html = ""
            for medicine in upcoming:
                upcoming_count += 1
                upcoming_card_html += card(medicine.title(), upcoming[medicine])

            completed_count = 0
```

```
            completed_card_html = ""
            for medicine in completed:
                completed_count += 1
                completed_card_html += card(medicine.title(), completed[medicine])
            session['upcoming_count'] = upcoming_count
            return render_template('app/schedule.html', completed_card_html=completed_card_html,
                                   upcoming_card_html=upcoming_card_html, upcoming_count=upcoming_count,
                                   completed_count=completed_count)
        else:
            return redirect('/login')
    except KeyError:
        return redirect('/login')


@app.route('/schedule/add', methods=["GET", "POST"])
def add_schedule():
    if session['user']:
        try:
            upcoming_count = session['upcoming_count']
            return render_template('app/add-medicine.html', upcoming_count=upcoming_count)
        except KeyError:
            return render_template('app/add-medicine.html')
    else:
        return redirect('/login')


@app.route('/schedule/submit', methods=['GET', "POST"])
def submit_schedule():
    data = request.form
    data = data.copy()
    print(data)
```

```
    dose_time = datetime.strptime(f"{data['hours']}:{data['minutes']} {data['halftime']}", '%I:%M %p')
    schedule_data = {
        'medicine_name': data['medicine-name'],
        'dose_time': dose_time.strftime("%H:%M:%S"),
        'start_date': datetime.strptime(data['start-date'].replace('-', ''), '%Y%m%d'),
        'end_date': datetime.strptime(data['end-date'].replace('-', ''), '%Y%m%d')
    }

    if schedule_data['start_date'] < schedule_data['end_date']:
        data['medicine-name'] = None
        data['hours'] = None
        data['minutes'] = None
        data['halftime'] = None
        data['start-date'] = None
        data['end-date'] = None

        days = []
        for key in data:
            if data[key]:
                days.append(key)
        schedule_data['notification'] = days
        add_medicine(session['user'], schedule_data)
        return redirect('/schedule')
    else:
        flash("Start date can't be before end date")
        return redirect('/schedule/add')


@app.route('/prescription')
def prescription_view():
    if session['user']:
```

```
        data, count = fetch_prescriptions(session['user'])
        prescription_card_html = ''
        for index in range(len(data)):
            prescription_card_html += prescription_card(data[index]['name'], data[index]['link'])
        try:
            upcoming_count = session['upcoming_count']
            return render_template('app/prescriptions.html', upcoming_count=upcoming_count,
total=count,
                                   prescription_card_html=prescription_card_html)
        except KeyError:
            return render_template('app/prescriptions.html', total=count,
prescription_card_html=prescription_card_html)
    else:
        return redirect('/login')



@app.route('/prescription/add', methods=["GET", "POST",'PUT'])
def add_prescription_page():
    if request.method == 'POST':
        try:
            data = request.form
            prescription_title = data.get('prescription-name')
            file = request.files['file']
            filename = secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
            path = f"{app.config['UPLOAD_FOLDER']}/{filename}"
            session['filename'] = path
            link = add_prescription(path, session['user'], filename)
            add_prescription_record(session['user'], prescription_title.title(), link)
            return redirect('/prescription')
        except:
            try:
```

```
            os.remove(session['filename'])
        except KeyError:
            pass
        return redirect('/prescription/add')
    try:
        upcoming_count = session['upcoming_count']
        return render_template('app/add-prescription.html', upcoming_count=upcoming_count)
    except KeyError:
        return render_template('app/add-prescription.html')



@app.route('/medicines')
def medicines():
    try:
        if session['user']:
            medicines = get_all_medicines(session['user'])
            medicine_card_html = ''
            total = 0

            for medicine in medicines:
                results = get_medicines(medicine)
                count = 0
                if results:
                    for index in range(len(results['names'])):
                        count += 1
                        if count < 3:
                            total += 1
                            try:
                                medicine_card_html += medicine_card(results['names'][index],
results['prices'][index],
                                                                    results['order links'][index])
```

```
                except IndexError:
                    pass
        try:
            upcoming_count = session['upcoming_count']
            return render_template('app/medicines.html', medicine_card_html=medicine_card_html,
total=total,
                                   upcoming_count=upcoming_count)

        except:
            return render_template('app/medicines.html', medicine_card_html=medicine_card_html,
total=total)

    else:
        return redirect('/login')
except KeyError:
    return redirect('/login')


@app.errorhandler(404)
def error(error):
    return '<h1>Error 404</h1>'


@app.errorhandler(500)
def error(error):
    return '<h1>Error 500</h1>'


@app.errorhandler(502)
def error(error):
    return '<h1>error502</h1>'
```

```
if __name__ == '__main__':
    app.run(debug=True)
```