002297666 Matthew Kuo

I am in charge of developing and deploying PAWS.

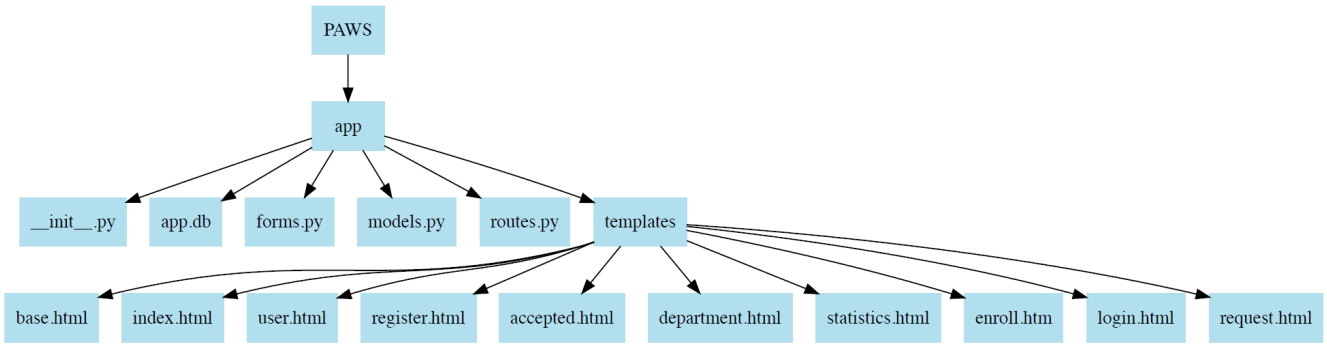
I have completed the following components:

1. Student Register/Login
2. Student Choose a semester
3. Student Add/Drop courses
4. Student View own class schedule
5. Administrator requests new ACCEPTED students from SLATE (you may hard code "GSU" in this request) (you need not provide a login for the administrator)
6. Administrator requests university level statistics (you may hard code "GSU"; Allow user to choose term and year from pull down list; no login required for this option as well)

Paws should also provide the following REST Web Services:

1. Given a department, return a list of students for the department.
2. Given a department, return a list of courses for the department.
3. Given department, return a list of enroll information for the department.

Structure of the website:



PAWS have 6 major components to handle different features and functionalities.

1. __init__.py

```python
from flask import Flask
from config import Config
from flask_sqlalchemy import SQLAlchemy
from flask_migrate import Migrate
from flask_login import LoginManager

app = Flask(__name__)
app.config.from_object(Config)
db = SQLAlchemy(app)
migrate = Migrate(app, db)
login = LoginManager(app)
login.login_view = 'login'

from app import routes, models

```

I use this file to keep track of some initial value and path of the whole web application.

2. app.db
   This is the SQLite database where all the data being used is stored. There are 4 main tables.
   a. course
      Course information is stored here.

id	name	time1	time2	department	student	semester	year

   b. departments
      It records how many departments we have at GSU.

id	department

   c. enroll_list
      It remembers which students enroll which course.

id	sid	cid

   d. user
      This table stores all information regarding each student.

id	username	email	password_hash	fname	lname	address1	address2	city	state	zip	degree	department

3. forms.py
   This python program creates object for forms that will be used in PAWS.
   There are:
   a. LoginForm
      In this form, there are 4 fields:

username	password	remember_me	submit

   b. RegistrationForm
      In this form, there are 13 fields:

username	email	password_hash	fname	lname	address1	address2	city	state	zip	degree	department	submit

   c. EnrollForm
      In this form, there are 2 fields

course	submit

   d. DepartmentForm
      In this form, there are 4 fields

department	semester	year	submit

4. models.py
   This program enables SQLAlchemy to know the data format in database.

```
1   from datetime import datetime
2   from app import db
3   from werkzeug.security import generate_password_hash, check_password_hash
4   from flask_login import UserMixin
5   from app import login
6
7   class User(UserMixin, db.Model): ...
31  class Course(db.Model): ...
42  class Departments(db.Model): ...
48  class EnrollList(UserMixin, db.Model): ...
55  @login.user_loader
56  def load_user(id):
57      return User.query.get(int(id))
58
```

5. routes.py
   This program controls every functions and routing methods for PAWS. It provides methods and api Including querying from database to show information for the website, letting outside source to receive certain information regarding students, departments, or semesters.

```
1   from flask import render_template, jsonify, abort, make_response, flash, redirect, url_for, request
2   from app import app, db
3   from app.forms import LoginForm, RegistrationForm, EnrollForm, DepartmentForm
4   from flask_login import current_user, login_user, logout_user, login_required
5   from werkzeug.urls import url_parse
6   from app.models import User, Course, EnrollList, Departments
7   import sqlite3 as sql
8   import requests
9
10  @app.route('/')
11  @app.route('/index')
12  @login_required
13  def index(): ...
16  @app.route('/login', methods=['GET', 'POST'])
17  def login(): ...
33  @app.route('/logout')
34  def logout(): ...
38  @app.route('/register', methods=['GET', 'POST'])
39  def register(): ...
52  @app.route('/user')
53  @login_required
54  def user(): ...
58  @app.route('/enroll', methods=['GET', 'POST'])
59  @login_required
60  def enroll(): ...
120 @app.route('/department', methods=['GET', 'POST'])
121 def department(): ...
160 @app.route('/api/get_courses/<string:dep>', methods=['GET'])
161 def get_courses(dep): ...
171 @app.route('/api/get_students/<string:dep>', methods=['GET'])
172 def get_students(dep): ...
182 @app.route('/api/get_enrollment/<string:dep>', methods=['GET'])
183 def get_enrollment(dep): ...
206 @app.route('/accepted', methods=['GET', 'POST'])
207 def accepted(): ...
216 @app.route('/statistics', methods=['GET', 'POST'])
217 def statistics(): ...
233
```

6. templates
   This folder contains all the html files that, as an interface, display information for or provide services to users.