Петрозаводский государственный университет
Институт математики и информационных технологий
Кафедра информатики и математического обеспечения

Направление подготовки бакалавриата

09.03.02 - Информационные системы и технологии

Отчет о практике по научно-исследовательской работе

Разработка и реализация подсистемы "Выпускники"веб-сервера Института математики и информационных технологий. База данных

Выполнила:
студентка 4 курса группы 22406

Т. А. Зинченко _____
подпись

Место прохождения практики:
Кафедра информатики и математического обеспечения

Период прохождения практики:
03.02.20-03.05.20

Руководитель:
О. Ю. Богоявленская к. т. н., доцент

_____
подпись

Итоговая оценка

_____
оценка

Петрозаводск — 2020

Содержание

Введение

Развитие web-технологий привело к появлению огромного количества сайтов с разной тематикой: форумы, блоги, информационные сайты, сайты крупных и малых фирм, государственных и образовательных учреждений и т.д.

Сегодня в сети Интернет представлена информация об Институте математики и информационных технологий (ИМиИТ) ПетрГУ. Сайт содержит всю необходимую информацию об учебном процессе, о кафедрах и ссылки на их страницы на официальном сайте ПетрГУ, новости и объявления института, информацию для абитуриентов. Целью проекта является предоставление в веб-пространстве информации о выпускниках ИМиИТ. Разработка раздела «Выпускники» сайта ИМиИТ позволит отслеживать карьерный рост выпустившихся студентов. Эта информация может помочь с выбором абитуриентам, которые хотят поступать в институт. Абитуриенты будут иметь четкое представление о будущей карьере после окончания института.

Целью работы является создание базы данных для подсистемы «Выпускники».

Первая версия подсистемы разрабатывалась в рамках дисциплины "Технология производства программного обеспечения"командой разработчиков: Т. А. Зинченко, Е. Ю. Коробкова, Д. А. Попова, Э. А. Сарконен, М. И. Фролова. Затем версия дорабатывалась.

Вся необходимая документация расположена на странице https://se.cs.petrsu.ru/wiki/GR.

Для достижения цели, поставленной выше, требуется выполнить следующие задачи:

- Для разработки системы нужно изучить следующие инструменты разработки: веб-фреймворк Flask, расширение Flask-SQLAlchemy и язык программирования Python.
- Выявить основные информационные объекты будущей базы данных. Определить связи между объектами и типы полей таблиц БД.
- С помощью выбранных инструментов разработать код базы данных.
- Для работы системы с базой данных необходимо установить подключение к БД.
- Необходимо создать функции для работы системы с БД: добавления,чтения,редактирования и удаления данных; поиск и сортировку данных; загрузку файлов.
- Разработать личный кабинет выпускника.
- Протестировать систему и выявить ошибки в работе системы.

1 Проектирование подсистемы "База данных"

1.1 Инфологическая модель

База данных будет содержать информацию о выпускниках: об обучении в университете, карьере. Для построения инфологической модели были спроектированы 5 информационных объектов: «Выпускник», «Мультимедиа», «Социальные сети», «Карьера» и «Администратор». Для наглядности используем диаграмму сущность-связь (ER-диаграмма). Для таких диаграмм характерно графическое изображение сущностей предметной области, их свойств и взаимосвязей между объектами. Сущностями являются наборы объектов, которые объединены в одну группу по набору одинаковых характеристик. Свойство или атрибут сущности - это характеристика сущности. Связь между сущностями - это ассоциация, позволяющая по одной сущности находить другие, связанные с ней.

Объект «Выпускник» предназначен для хранения информации о выпускнике и имеет следующие свойства: ФИО, Логин, Пароль, Дата рождения, Семейное положение, Email, Кафедра, Специальность/Направление, Год выпуска, Научный руководитель, Тема работы.

Рис. 1: Объект "Выпускник"

Объект «Мультимедиа» предназначен для хранения фото- и видеоматериалов выпускника, имеет следующие свойства: Имя мультимедиа файла, Дата загрузки, Подпись, Лайки.

Рис. 2: Объект "Мультимедиа"

Объект «Социальные сети» предназначен для хранения ссылок на социальные сети выпускника, имеет следующие свойства: Ссылка на социальную сеть.

Рис. 3: Объект "Социальные сети"

Объект «Карьера» предназначен для информации о карьере выпускника, имеет следующие свойства: Сфера работы, Место работы, Должность, Период работы.

Рис. 4: Объект "Карьера"

Объект «Администратор» предназначен для хранения логинов и паролей администраторов, имеет следующие свойства: Логин, Пароль.

Рис. 5: Объект "Администратор"

Связь один-ко-многим обозначает, что одному представителю объекта соответствуют несколько представителей другого объекта. Поэтому объект «Выпускник» связан отношением один-ко-многим с объектами «Карьера», «Социальные сети» и «Мультимедиа», так как выпускник может указывать несколько мест работ, социальных сетей и загружать много мультимедиа файлов.

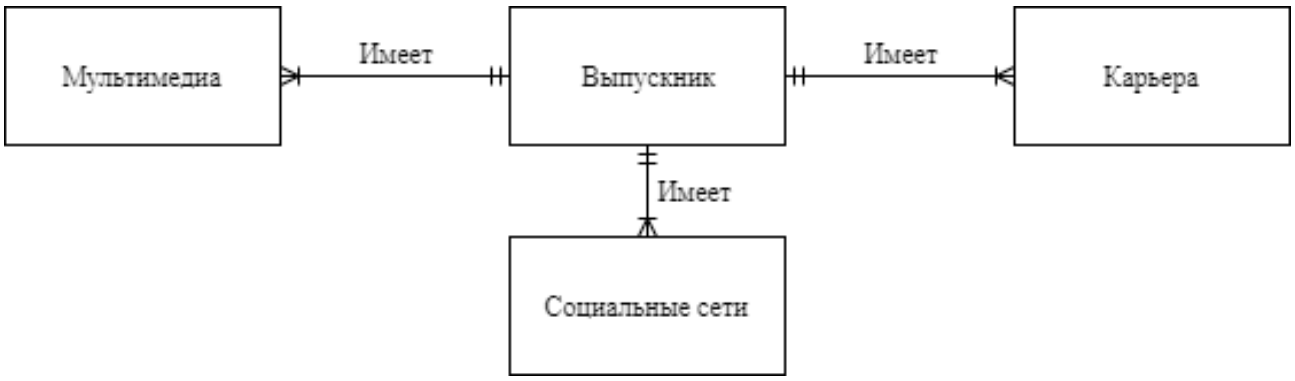

Рис. 6: Инфологическая модель

1.2 Схема базы данных

По инфологической модели была построена схема базы данных на основе реляционной модели данных.

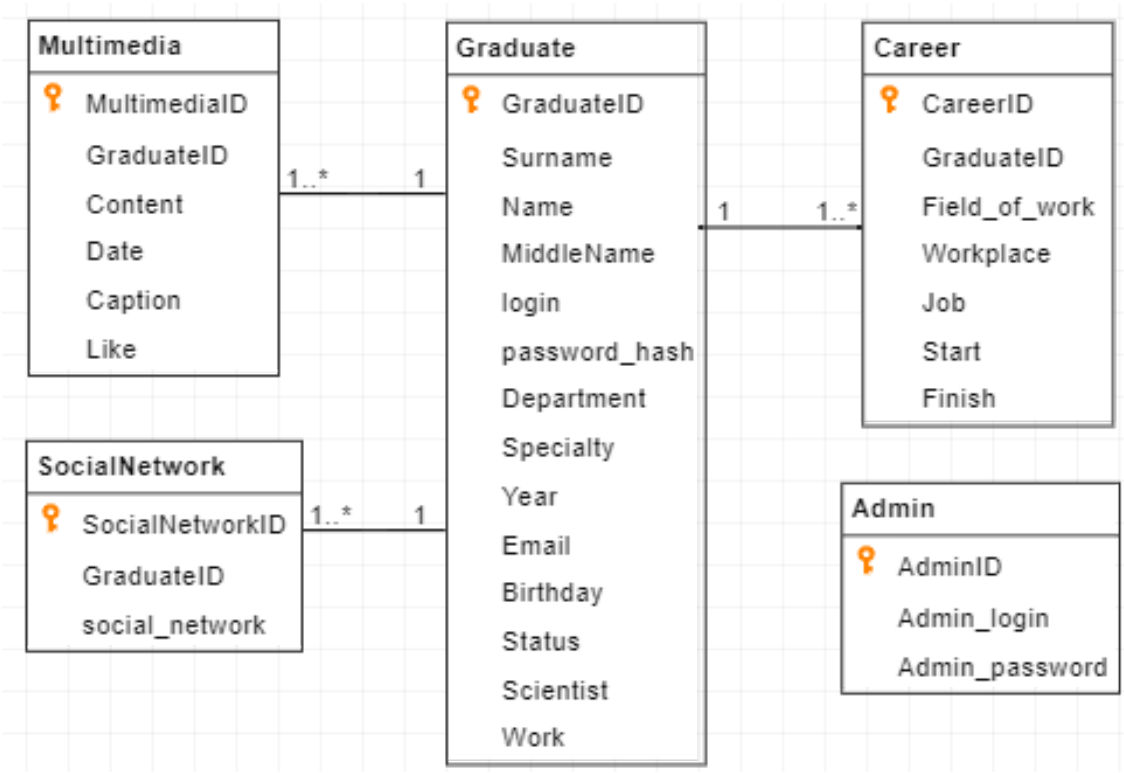

Рис. 7: Схема базы данных

Класс «Выпускник» предназначен для информации о выпускнике и включает следующие поля:

Таблица 1: Выпускник

Поле	Ключ	Тип
Идентификатор выпускника	PK	integer
Фамилия		string
Имя		string
Отчество		string
Логин		string
Пароль		string, хранит хеш пароля
Кафедра		string
Специальность/Направление		string
Год выпуска		string
Email		string
Дата рождения		DateTime
Семейное положение		string
Научный руководитель		string
Тема работы		string

Класс «Карьера» предназначен для информации о карьере выпускника после окончания университета, имеет следующие поля:

Таблица 2: Карьера

Поле	Ключ	Тип
Идентификатор работы	PK	integer
Идентификатор выпускника	FK	integer
Сфера работы		string
Место работы		string
Должность		string
Начало периода работы		DateTime
Конец периода работы		DateTime

Класс «Мультимедиа» предназначен для мультимедиа файлов выпускника и включает в себя:

Таблица 3: Мультимедиа

Поле	Ключ	Тип
Идентификатор	PK	integer
Идентификатор выпускника	FK	integer
Имя мультимедиа файла		string
Дата загрузки		DateTime
Подпись		string
Количество лайков		integer

Класс «Социальные сети» предназначен для социальных сетей выпускников и включает в себя:

Таблица 4: Социальные сети

Поле	Ключ	Тип
Идентификатор	PK	integer
Идентификатор выпускника	FK	integer
Ссылка на социальную сеть		string

Класс «Администратор» предназначен для логинов и паролей администраторов:

Таблица 5: Администратор

Поле	Ключ	Тип
Идентификатор администратора	PK	integer
Логин		string
Хеш пароля		string

2 Реализация подсистемы "База данных"

2.1 Инструменты разработки

Для реализации системы «Выпускники» используется веб-фреймворк Flask. Flask – это веб-фреймворк, написанный на языке программирования Python, для создания веб-приложений. «Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода» [6]

В разработке используем базу данных SQLite — это встраиваемая реляционная база данных, которая представляет собой один файл. Для создания базы данных используется расширение Flask-SQLAlchemy. «SQLAlchemy — это программная библиотека на языке Python для работы с реляционными СУБД. Служит для синхронизации объектов Python и записей реляционной базы данных.

SQLAlchemy позволяет описывать структуры баз данных и способы взаимодействия с ними на языке Python без использования SQL» [5]. Хранящиеся в базе данные представляют наборами классов. Для определения столбцов используют атрибут Column. Первичные ключи помечаются с помощью аргумента primary_key = True. Несколько ключей можно пометить как первичные ключи, в таком случае они становятся составным первичным ключом. Первым аргументом Column является тип столбца. Наиболее распространенные типы:

Таблица 6: Типы данных

Тип	Описание
Integer	Целое число
String(size)	Строка с максимальной длиной
Text	Длинный текст
DateTime	Дата и время
Float	Число с плавающей точкой
Boolean	Логическое значение
LargeBinary	Большие произвольные двоичные данные

Отношения выражаются с помощью функции relationship (). Первичный ключ обозначается primary_key = True. Внешний ключ обозначется через ForeignKey (‘имя столбца’). Пример создания отношений:

```
class Person(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(50), nullable=False)
    addresses = db.relationship('Address', backref='person', lazy=True)

class Address(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.String(120), nullable=False)
    person_id = db.Column(db.Integer, db.ForeignKey('person.id'),
        nullable=False)
```

Рис. 8: Пример создания отношений

В классе Address для person_id указан внешний ключ person.id, где person – название таблицы, id – столбец. В классе Person указано отношение на класс Address, где backref – ссылка на таблицу person.

2.2 Реализация

С помощь программной библиотеки SQLAlchemy реализуем построенную схему базы данных в файловой системе. Пример кода для класса «Карьера» приведен в Приложении

Для подключения системы «Выпускники» к базе данных ~~нужно~~ в файле конфигурации указать местоположение базы данных: SQLALCHEMY_DATABASE_URI = ’sqlite:///’ + os.path.join(basedir, ’app.db’).

Для работы с базой данных были созданы функции:

- def sort() - функция для сортировки по заданным критериям. Входные данные: год выпуска, кафедра, направление или сфера работы.
- def search() – функция поиска выпускника по фамилии. Входные данные: подстрока поиска.
- def add_<имя класса>() – функция добавления информации для каждого классы БД. Входные данные: данные из форм html.
- def uploaded_file (filename) – функция загрузки файлов мультимедиа в каталог системы. Входные данные: файл мультимедиа.
- def del_<имя класса>() – функция удаление записей для каждого класса. Входные данные: идентификатор записи.
- def update_<имя класса>() – функция редактирования записей для каждого класса. Входные данные: идентификатор записи и данные из форм html.

Пример кода функции def search() приведен в Приложении Б.

Декоратор @app.route - создает связь между URL-адресом (аргумент) и функцией. Когда браузер будет запрашивать один из этих адресов, Flask будет вызывать соответствующую функцию и передавать возвращаемое значение обратно в качестве ответа или перенаправлять на другие страницы. Эти функции принимают данные из форм, добавляют их в таблицы и перенаправляют на страницы с выводом этих данных.

2.3 Разработка личного кабинета

Личный кабинет выпускника предоставляет возможность просмотреть общий список выпускников, выполнить поиск информации о выпускниках по критериям, добавить мультимедиа файлы (фото и видео), добавить или отредактировать информацию о себе.

Для разработки личного кабинета ~~использ~~уем расширение Flask-Login, которое обеспечивает управление сеансами пользователя для Flask. Расширение обрабатывает общие задачи входа в систему, выхода из системы и запоминания сеансов пользователей в течение длительного периода времени.

Создан менеджер входа: login_manager = LoginManager(). Он содержит код, который позволяет приложению и Flask-Login работать вместе. Далее он настроен на вход в систему

с помощью login_manager.init_app(app). Реализована функция авторизации выпускника в системе.

Также реализованы вывод информации о выпускнике из базы данных, возможность добавления и редактирования информации выпускником, загрузка мультимедиа файлов.

3 Тестирование системы

Разработан документ тестирования со следующими видами тестирования:

- Аттестационное тестирование - тестирование системы, которое определяет соответствие требованиям, на которых основана система.
- Блочное тестирование - тестирование отдельных модулей, которое выполняется отдельно от других частей сиситемы.
- Интеграционное тестирование - тестирование подсистем и их взаимодействия между собой.

Заключение

Получены следующие результаты:

- Установлены все необходимые инструменты для разработки
- Изучены инструменты разработки
- Спроектирована схема базы данных
- Реализована схема базы данных
- Прототип системы «Выпускники» подключен к базе данных
- Организована работа системы с базой данных
- Ведется разработка личного кабинета
- Запланировано тестирование системы

Список литературы

1. Документация Flask-SQLAlchemy [Электронный ресурс]

   Режим доступа: http://flask-sqlalchemy.pocoo.org/

   (дата обращения: 30.04.2019)

2. Документация Flask [Электронный ресурс]

   Режим доступа: http://flask.pocoo.org

   (дата обращения: 15.03.2019)

3. Самоучитель Python [Электронный ресурс]

   Режим доступа: https://pythonworld.ru/samouchitel-python

   (дата обращения: 15.03.2019)

4. SQLAlchemy [Электронный ресурс]

   Режим доступа: https://ru.wikipedia.org/wiki/SQLAlchemy

   (дата обращения: 25.11.2019)

5. Python [Электронный ресурс]

   Режим доступа: https://ru.wikipedia.org/wiki/Python

   (дата обращения: 25.11.2019)

6. Проектирование баз данных [Электронный ресурс]

   Режим доступа: https://ru.wikipedia.org/wiki/Проектирование_баз_данных

   (дата обращения: 03.03.2019)

7. Документация Flask Uploads [Электронный ресурс]

   Режим доступа: http://flask.pocoo.org/docs/0.12/patterns/fileuploads/

   (дата обращения: 06.03.2019)

8. Документация SQLAlchemy [Электронный ресурс]

   Режим доступа: https://docs.sqlalchemy.org/en/latest/core/type_basics.html

   (дата обращения: 06.03.2019)

9. Загрузка файлов [Электронный ресурс]

Режим доступа: https://flask-russian-docs.readthedocs.io/ru/latest/patterns/fileuploads.html

(дата обращения: 16.03.2019)

10. Flask-Uploads [Электронный ресурс]

Режим доступа: https://flask-uploads.readthedocs.io/en/latest/configuration

(дата обращения: 16.03.2019)

11. Примеры использования модуля os в Python [Электронный ресурс]

Режим доступа: https://python-scripts.com/import-os-example

(дата обращения: 13.04.2019)

12. Мега-Учебник Flask, Часть 6: Страница профиля и аватары (издание 2018) [Электронный ресурс]

Режим доступа: https://habr.com/ru/post/346348/

(дата обращения: 13.04.2019)

13. ЦИФРОВЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ, НАУКЕ, ОБЩЕСТВЕ Материалы XIII всероссийской научно-практической конференции [Электронный ресурс]

Режим доступа: https://it2019.petrsu.ru/doc/it2019.pdf

(дата обращения: 01.12.2019)

14. Flask-Login [Электронный ресурс]

Режим доступа: https://flask-login.readthedocs.io/en/latest/

(дата обращения: 20.04.2020)

Приложение

Приложение А

Листинг 1: Код для класса"Карьера"

```
class Career(db.Model):
        __tablename__ = 'career'
        CareerID = db.Column(db.Integer, primary_key=True)
        GraduateID = db.Column(db.Integer,
                db.ForeignKey('graduate.GraduateID',
                        ondelete='CASCADE'))
        Field_of_work = db.Column(db.String(1024))
        Workplace = db.Column(db.String(1024))
        Job = db.Column(db.String(1024))
        Start = db.Column(db.DateTime())
        Finish = db.Column(db.DateTime())
        graduate_id = db.relationship(Graduate, primaryjoin=
          ( GraduateID == Graduate.GraduateID ),
                backref=backref('graduates', cascade='delete'))

        def __init__(self, GraduateID, Field_of_work,
                        Workplace, Job, Start, Finish):
                self.GraduateID = GraduateID
                self.Field_of_work = Field_of_work
                self.Workplace = Workplace
                self.Job = Job
                self.Start = Start
                self.Finish = Finish
```

Приложение Б

Листинг 2: Функция поиска по фамилии

```
@app.route('/search', methods=['POST'])
def search():
        if request.form['search'] != "":
                search = request.form['search']
```

```
            search = search.capitalize()
    else:
            search = Graduate.Surname
    graduates=db.session.query(Graduate.GraduateID,
            Graduate.Surname, Graduate.Name,
            Graduate.MiddleName, Graduate.login,
            Graduate.passwd, Graduate.Department,
            Graduate.Specialty, Graduate.Year,
            Graduate.Email, Graduate.Birthday,
            Graduate.Status, Graduate.Scientist,
            Graduate.Work, Career.Field_of_work,
            Career.Workplace, Career.Job, Career.Start,
            Career.Finish).filter(Graduate.GraduateID ==
                    Career.GraduateID,
            Graduate.Surname.like('%'+search+'%')).all()
    return render_template('sort_html.html',
            graduates=graduates)
```