

## ГЛАВА 1

### Постановка задачи

#### 1.1 Цель работы

Освоить навыки верстки с использованием HTML5, CSS3, JavaScript; навыки разработки серверной части веб-приложения на языке программирования, построением шаблонов.

#### 1.2 Индивидуальное задание

Разработка хранилища учебных материалов.

Необходима поддержка различных разрешений экрана, в том числе мобильных. Страницы должны генерироваться по запросу пользователя с использованием шаблонов.

## ГЛАВА 2

### Общее описание приложения

#### 2.1 Используемые инструменты


В разработке сайта был использован язык программирования Python и микрофреймворк Flask, который отвечал за обработку запросов клиента. Также в состав данного фреймворка входит шаблонизатор Jinja2.

Также использовался CSS-препроцессор - SASS. Он позволяет соблюдать принцип DRY (англ. Don't repeat yourself - не повторять свой код), предоставляя расширенные возможности, отсутствующие в CSS, а именно:

- переменные - существует возможность один раз задать значение переменной, затем многократно использовать в коде;
- миксины - "функции" языка, сохранив целые части кода, можно использовать их многократно, в том числе с использованием переменных. Также существует ряд готовых библиотек (в данной работе использовалась библиотека Bourbon);
- вложенные стили и наследование - они позволяют использовать правила одного элемента в другом.

#### 2.2 Файловая структура приложения

Файловая структура веб-приложения имеет следующий вид:



`codes/file_struct.png`

Запуск сервера осуществляется с помощью файла `run.py`:

После запуска сервера приложения, оно становится доступно по адресу: `http://localhost:5000`. Для того, чтобы сделать сервер доступным в сети или включить режим отладки, необходимо внести следующие строки в файл `config.py`:

Каждая страница сайта формируется при запросе пользователя, шаблоны находятся по адресу `/public_html/app/templates/`. Общий шаблон для всех страниц `base.html` отвечает за отображение панели навигации, бокового меню, контейнера с содержимым страницы и футера.

Главная страница сформирована из шаблона `/public_html/app/templates/index.html` в котором содержатся шаблоны "карточек дисциплин" равным количеством дисциплин, зарегистрированных в приложении.

Информация, для отображения страницы "About" находится в фай-

ле about.html. Тексты лекций находятся в соответствующих подкаталогах каталога /templates. Также существует возможность загрузки pdf-файла каждой лекции, которые находятся в директории /static. Также в этой папке находятся файлы, отвечающие за оформление страниц (/\_css/main.css) и за интерактивное поведение элементов страниц (/\_js/main.js). В каталоге /fonts расположены файлы гарнитур, используемые на страницах сайта.

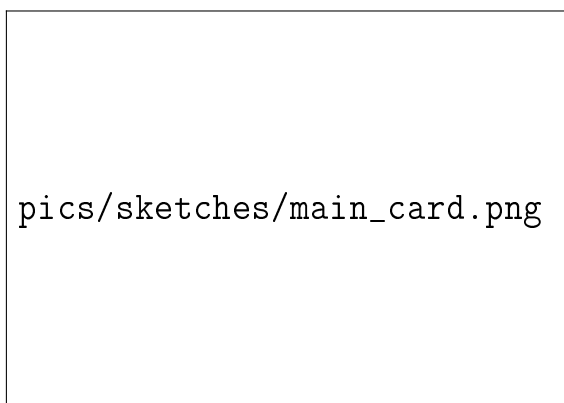
В каталогах /\_sass и /bourbon находятся неcompiled файлы стилей и библиотека готовых стилей соответственно.

## ГЛАВА 3

### Макет интерфейса приложения

Перед разработкой приложения был составлен макет интерфейса. На нем отражены следующие составляющие сайта:

- элементы приложения, их вид и расположение;
- вид приложения при разных разрешениях экрана;
- цветовое решение;
- графические элементы;
- гарнитуры и шрифт текстов.



(а) Дисциплины в виде карточек



(б) Дисциплины в виде списка

Рисунок 3.1 — Главная страница приложения

## ГЛАВА 4

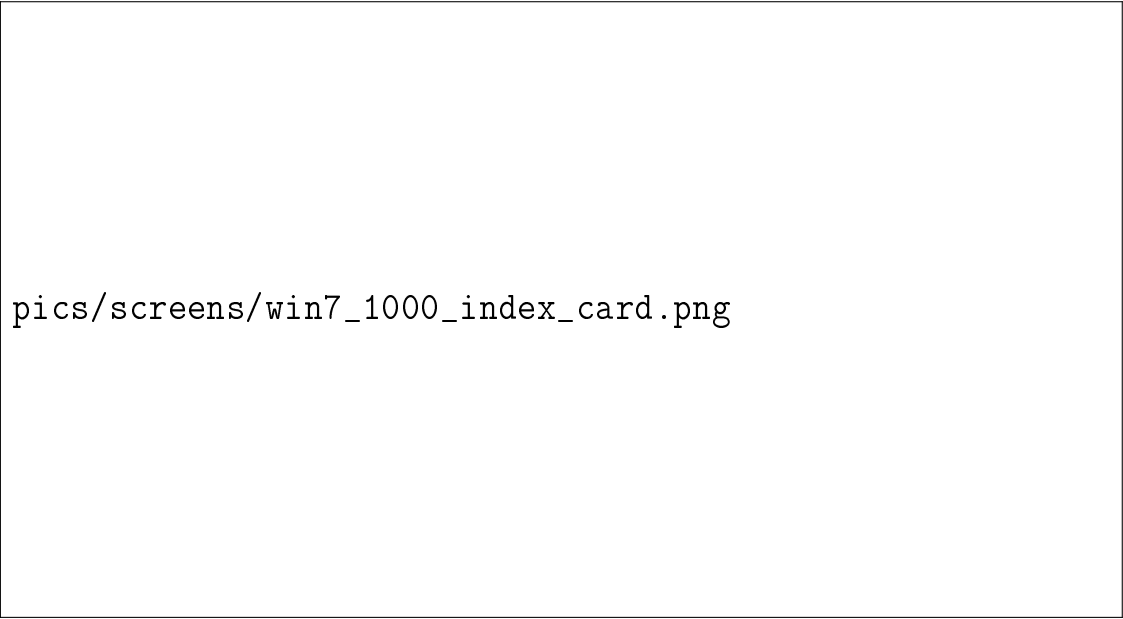
### Скриншоты приложения

Приложение оптимизированно для разных разрешений экрана (больше 900, 700, 480 пикселей по ширине) а так же версия для печати. Для этого использовались @media-правила, встроенные в CSS.

Приложение протестировано в браузере Google Chrome на следующих операционных системах:

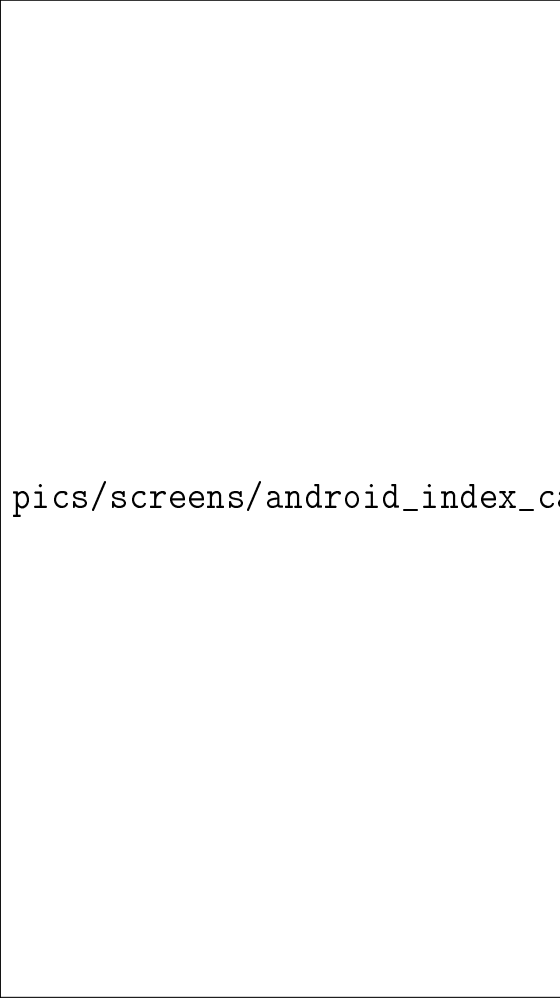
- Windows 7 x64, версия 43.0.2357.81
- Mac OS X, версия 43.0.2357.81
- Android 4.4.4, версия 43.0.2357.92
- iOS 8.3, версия 42.0.2311.47

Как можно увидеть на рисунках, готовое приложение выглядит также, как и проектировалось на макетах. Небольшие различия объясняются итеративным подходом к разработке - некоторые изменения вносились уже на этапе разработки.



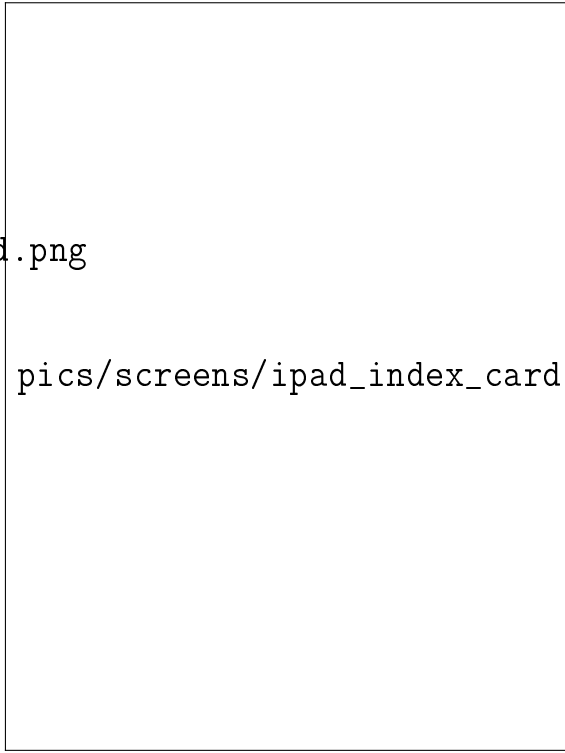
pics/screens/win7\_1000\_index\_card.png

(a) Windows 7



pics/screens/android\_index\_card.png

(б) Android



pics/screens/ipad\_index\_card.png

(в) iOS

Рисунок 4.1 — Главная страница с дисциплинами в виде карточек

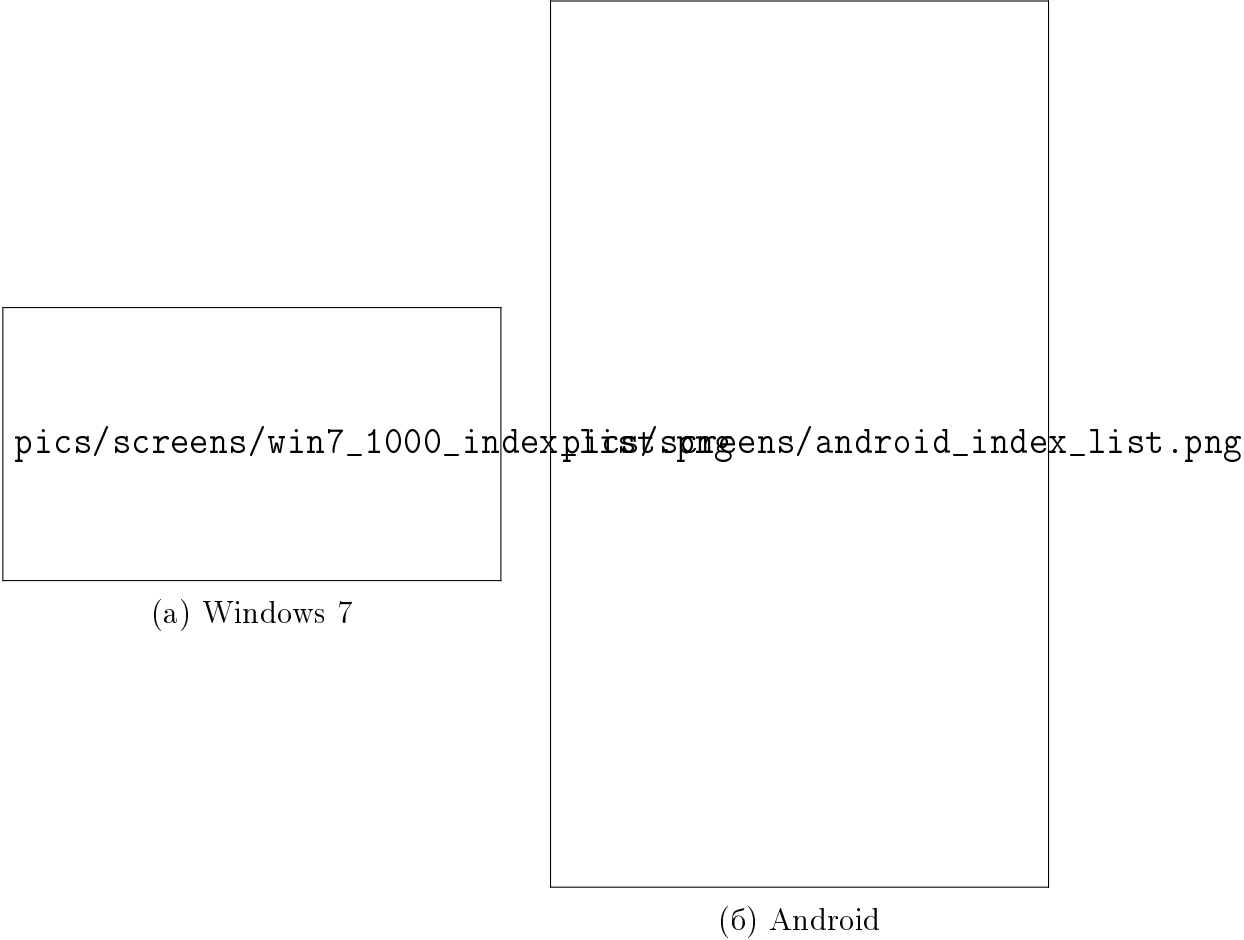


Рисунок 4.2 — Главная страница с дисциплинами в виде списка

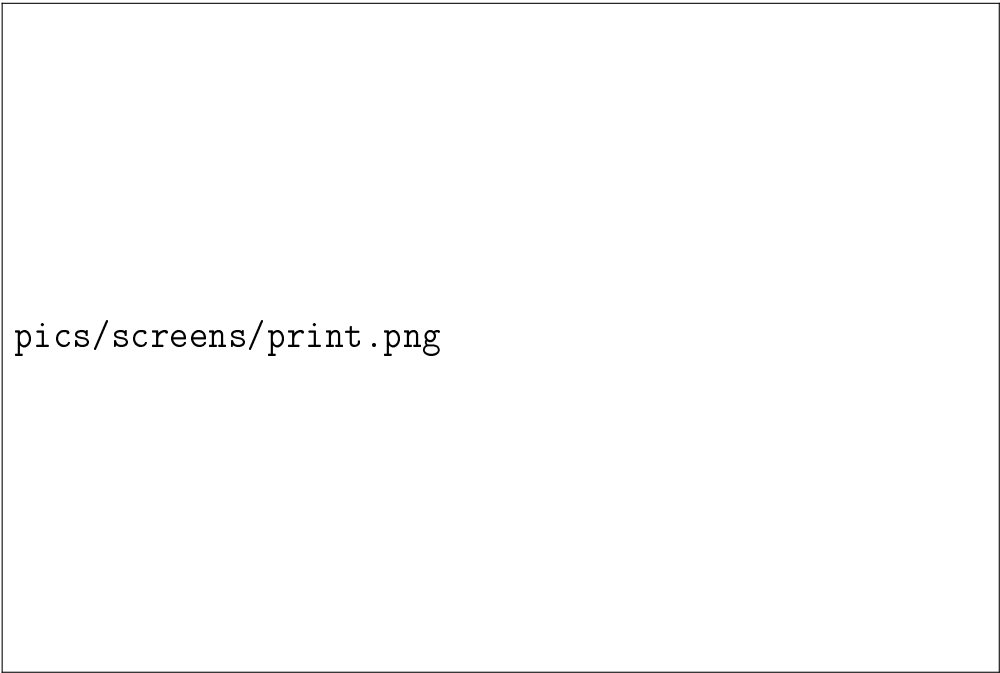
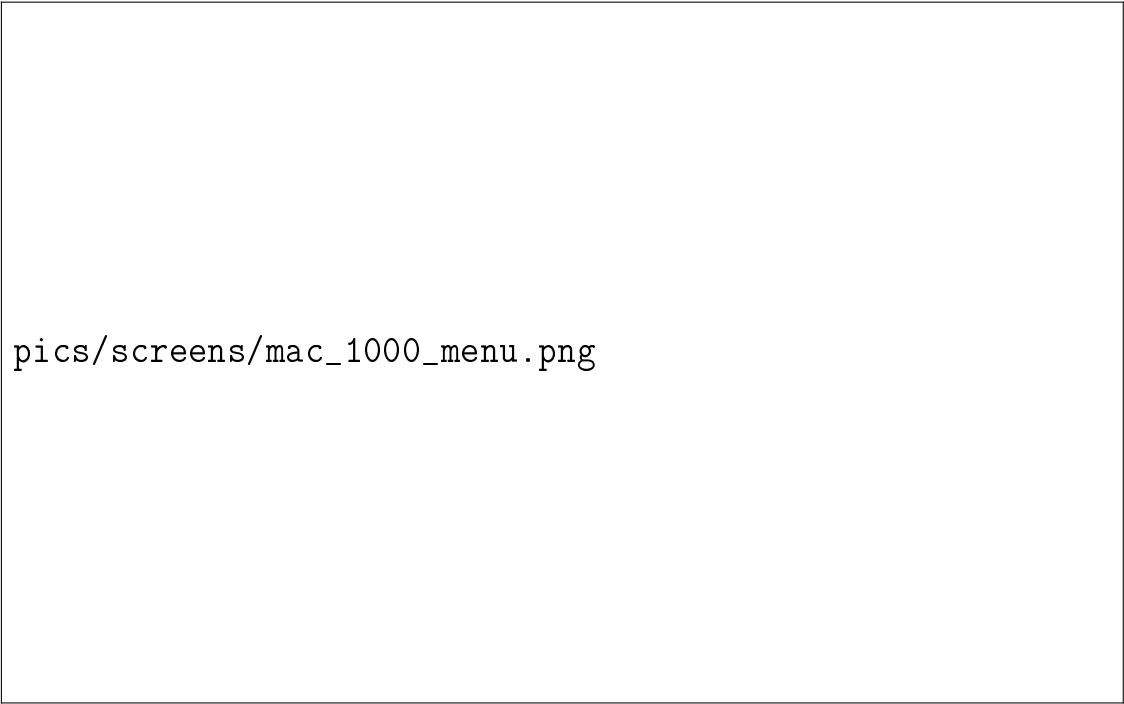


Рисунок 4.3 — Версия для печати





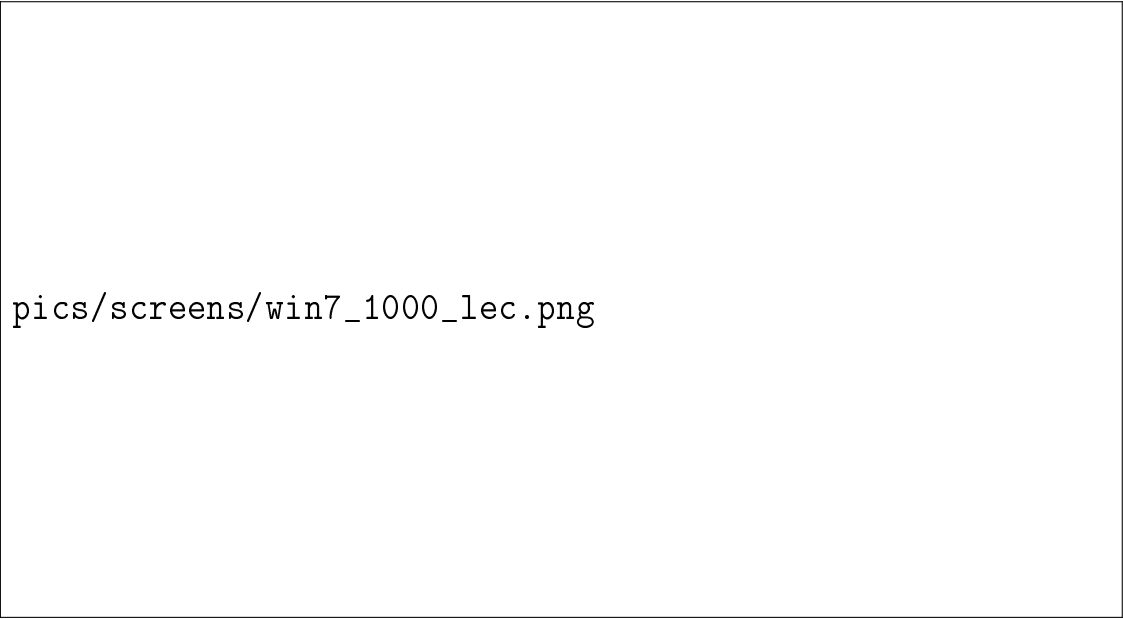
(a) Mac OS X



(б) Android

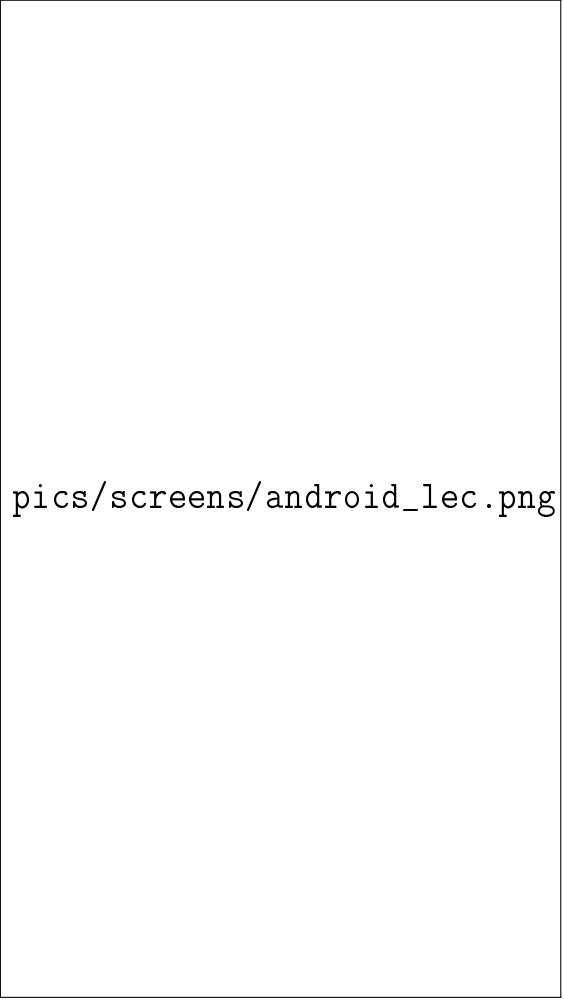
(в) iOS

Рисунок 4.4 — Меню с разными разрешениями экранов



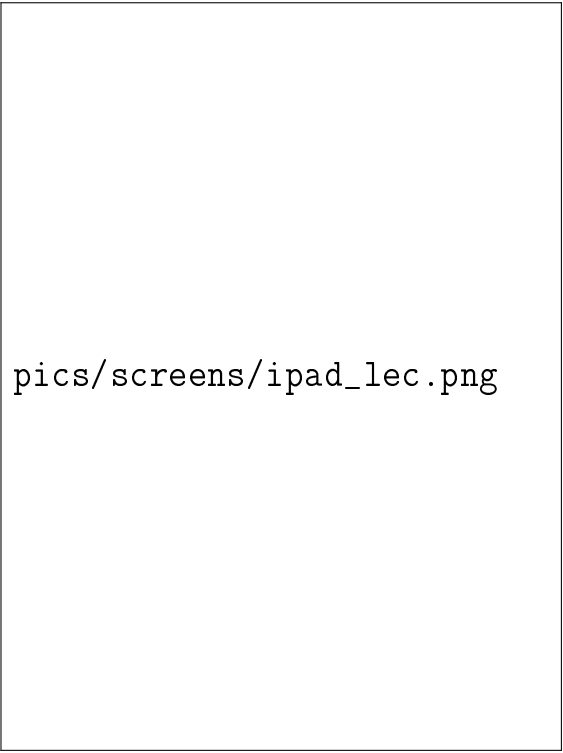
pics/screens/win7\_1000\_lec.png

(a) Windows 7



pics/screens/android\_lec.png

(б) Android



pics/screens/ipad\_lec.png

(в) iOS

Рисунок 4.5 — Текст лекции

## ГЛАВА 5

## Листинги приложения

Все исходные коды приложения можно разделить на серверную и клиентскую части. В серверную часть входят файлы, запускаемые на сервере и участвующие в процессе разработки, а в клиентскую - то, что отображается пользователю - шаблоны и стили.

```
1 from flask import Flask
2
3 app = Flask(__name__)
4 app.config.from_object('config')
5 from app import views
```

Листинг 5.1 — \_\_init\_\_.py

```
1 from flask import render_template
2 from flask import send_file
3 from app import app
4
5 class Discipline:
6     def __init__(self, code, name, pdf):
7         self.code = code
8         self.name = name
9         self.pdf = pdf
10
11 web = Discipline('web', 'Веб-
        программирование', 'webPDF')
12 ics = Discipline('ics', 'Проектирование
        АСОИУ', 'icsPDF')
13 electr = Discipline('electr', 'Схемотехника',
        'electrPDF')
14
15 discipline_list = [web, ics, electr]
16
17 @app.route('/')
18 @app.route('/index')
19 def index():
20     return render_template('index.html', discipline_list =
        discipline_list)
21
22 @app.route('/about')
```

```

23 def about():
24     return render_template('about.html', discipline_list =
                               discipline_list)
25
26 @app.route('/web')
27 def web():
28     return render_template('/web/index.html', discipline_list =
                               discipline_list)
29
30 @app.route('/webPDF')
31 def webPDF():
32     return send_file('static/pdfs/web.pdf', as_attachment=True)
33
34 @app.route('/ics')
35 def ics():
36     return render_template('/ics/index.html', discipline_list =
                               discipline_list)
37
38 @app.route('/icsPDF')
39 def icsPDF():
40     return send_file('static/pdfs/ics.pdf', as_attachment=True)
41
42 @app.route('/electr')
43 def electr():
44     return render_template('electr/index.html', discipline_list
                               = discipline_list)
45
46 @app.route('/electrPDF')
47 def electrPDF():
48     return send_file('static/pdfs/electr.pdf',
                       as_attachment=True)

```

Листинг 5.2 — views.py

```

1 from app import app
2
3 # app.run(host='192.168.0.199')
4 app.run()

```

```
5 # app.run(debug=True)
```

Листинг 5.3 — run.py

## 5.1 Шаблоны