**Часть А.**

1. **a) pip install flask**
2. **a) Flask()**
3. **c) @app.route("/home")**
4. **c) debug=True**
5. **b) request.args**
6. **c) render\_template()**
7. **c) flask run**
8. **a) <var>**
9. **a) 5000**

**Часть B.**

1. **Flask — это легковесный фреймворк для создания веб-приложений на языке Python. Он подходит для разработки как небольших сайтов, так и крупных веб-приложений, обеспечивая гибкость и простоту настройки.**
2. **Структура типичного Flask-проекта:**
   * **/static/ — папка для CSS, JS, изображений и других статических файлов.**
   * **/templates/ — папка для HTML-шаблонов.**
   * **/app/ — папка с маршрутами и логикой приложения.**
     1. **\_\_init\_\_.py – инциализатор приложения**
     2. **routes.py – файл с маршрутами(URL) приложения**
     3. **models.py – файл с моделями(необязателен)**
     4. **forms.py – файл с формами(необязателен)**
   * **config.py — конфигурационные параметры.**
   * **requirements.txt — список зависимостей для установки через pip.**
3. **Маршрутизация в Flask позволяет связывать URL-адреса с конкретными функциями. Например:**
4. *from* flask *import* Flask
5. app = Flask(\_\_name\_\_)
6. @app.route("/home")
7. def **home**():
8. *return* "Добро пожаловать на главную страницу!"
9. **Функция render\_template() используется для отображения HTML-шаблонов и передачи в них данных. Например:**
10. @app.route("/profile")
11. def **profile**():
12. *return* render\_template("profile.html", *username*="Иван")
13. **Контекст запроса в Flask обеспечивает доступ к данным о текущем HTTP-запросе. Например, получение аргументов из строки запроса:**
14. *from* flask *import* request
15. @app.route("/search")
16. def **search**():
17. query = request.args.get("q")
18. *return* f"Вы искали: {query}"
19. **Переменные окружения в Flask позволяют хранить конфиденциальные данные или параметры конфигурации за пределами кода. Например:**
20. *import* os
21. app.config['SECRET\_KEY'] = os.getenv('SECRET\_KEY')
22. **Перенаправление (redirect) в Flask используется для перенаправления пользователя на другой маршрут. Пример:**
23. *from* flask *import* redirect
24. @app.route("/old")
25. def **old**():
26. *return* redirect("/new")
27. **Конфигурация (config) в Flask хранит параметры, такие как ключи, настройки базы данных и режимы работы. Настраивается через объект конфигурации app.config. Пример:**
28. app.config['DEBUG'] = True

**Часть С.**

**Пример реализации задания:**

*from* flask *import* Flask, render\_template, request

*import* sqlite3

app = Flask(\_\_name\_\_)

*# Подключение к базе данных SQLite*

def **get\_db\_connection**():

    conn = sqlite3.connect('database.db')

    conn.row\_factory = sqlite3.Row

*return* conn

*# Маршрут для детальной информации о пользователе*

@app.route('/user/<int:id>')

def **user\_detail**(*id*):

    conn = get\_db\_connection()

    user = conn.execute('SELECT \* FROM users WHERE id = ?', (*id*,)).fetchone()

    conn.close()

*return* render\_template('user\_detail.html', *user*=user)

*# Маршрут для отображения списка пользователей*

@app.route('/users')

def **users**():

    conn = get\_db\_connection()

    users = conn.execute('SELECT \* FROM users').fetchall()

    conn.close()

*return* render\_template('users.html', *users*=users)

*# Загрузка статических файлов*

@app.route('/static/<path:filename>')

def **static\_files**(*filename*):

*return* app.send\_static\_file(*filename*)

*if* \_\_name\_\_ == "\_\_main\_\_":

    app.run(*debug*=True)