from flask import Flask, render\_template

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

# Определяем маршрут для главной страницы

@app.route('/')

def index():

return render\_template('index.html')

# Определяем маршрут для страницы 404

@app.errorhandler(404)

def page\_not\_found(error):

return render\_template('page404.html'), 404

# Запускаем приложение

if \_\_name\_\_ == '\_\_main\_\_':

app.run()

<!-- index.html -->

<!DOCTYPE html>

<html>

<head>

<title>Главная страница</title>

<link rel="stylesheet" href="{{ url\_for('static', filename='css/style.css') }}">

</head>

<body>

<h1>Добро пожаловать на мой сайт</h1>

<p>Здесь будет информация о компании или других страницах меню</p>

</body>

</html>

<!-- page404.html -->

<!DOCTYPE html>

<html>

<head>

<title>Страница не найдена</title>

</head>

<body>

<h1>Ошибка 404: Страница не найдена</h1>

<p>К сожалению, запрашиваемая страница не существует. Пожалуйста, проверьте правильность введенного адреса.</p>

</body>

</html>

/\* style.css \*/

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

color: #333;

}

h1 {

color: #007bff;

}

p {

font-size: 16px;

}

Sure, here is an example of a Flask project with four menu items related to travel:

1. Create a Python file named app.py with the following content:

from flask import Flask, render\_template

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

@app.route('/')

def home():

return render\_template('home.html')

@app.route('/about')

def about():

return render\_template('about.html')

@app.route('/destinations')

def destinations():

return render\_template('destinations.html')

@app.route('/blog')

def blog():

return render\_template('blog.html')

@app.errorhandler(404)

def page\_not\_found(e):

return render\_template('404.html'), 404

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

1. Create a folder named templates and within it, create six HTML files (home.html, about.html, destinations.html, blog.html, and 404.html) with the desired content.
2. Here is an example content for the HTML files:

* home.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Home</title>

</head>

<body>

<h1>Welcome to our travel website!</h1>

</body>

</html>

* about.html, destinations.html, and blog.html: You can create similar content as above but with different information for each page.
* 404.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Page Not Found</title>

</head>

<body>

<h1>404 - Page Not Found</h1>

</body>

</html>

1. Run the Flask application by executing python app.py in the terminal. You can access the website by visiting http://127.0.0.1:5000/ in your browser.

Now you have a simple Flask project with four menu items related to travel and a custom 404 error page.

Создайте новую папку для проекта и в ней создайте файл с расширением .py (например, app.py), который будет содержать основной код Flask приложения.

1. Установка и импорт необходимых модулей:

Установите Flask, если у вас его еще нет, используя pip install flask. Импортируйте модуль Flask в вашем файле .py:

from flask import Flask, render\_template

1. Определение маршрутов и функций представления:

Определите функции представления для каждой страницы и свяжите их с соответствующими URL-адресами. Например:

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/about')

def about():

return render\_template('about.html')

@app.route('/destinations')

def destinations():

return render\_template('destinations.html')

# Добавьте еще два маршрута для оставшихся страниц

1. Создание HTML страниц:

Создайте шесть HTML файлов для каждой страницы (index.html, about.html, destinations.html и т. д.) в папке templates внутри вашего проекта. Заполните их содержимым, соответствующим каждой странице вашего сайта о путешествиях.

1. Обработка ошибки 404:

Добавьте обработчик ошибки 404 в вашем приложении Flask, чтобы отображать специальную страницу с сообщением об ошибке. Например:

@app.errorhandler(404)

def page\_not\_found(error):

return render\_template('404.html'), 404

1. Запуск приложения:

Запустите ваше Flask приложение, запустив файл .py:

$ python app.py

После этого ваш сайт о путешествиях с четырьмя пунктами меню и обработкой ошибки 404 будет доступен по адресу [http://127.0.0.1:5000/.](http://127.0.0.1:5000/) В случае ошибки ввода несуществующего URL-адреса пользователь будет перенаправлен на страницу 404.html.

1. Создайте директорию проекта и поместите в нее один файл Python (например, app.py) и шесть файлов HTML (например, index.html, about.html, destinations.html, contact.html, not\_found.html).
2. В файле app.py подключите Flask и создайте объект приложения:

from flask import Flask, render\_template

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

# Создание пунктов меню

menu = {

'Главная': '/',

'О нас': '/about',

'Путеводитель': '/destinations',

'Контакты': '/contact',

}

# Создание маршрутов для каждой страницы

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/about')

def about():

return render\_template('about.html')

@app.route('/destinations')

def destinations():

return render\_template('destinations.html')

@app.route('/contact')

def contact():

return render\_template('contact.html')

# Обработка ошибки 404

@app.errorhandler(404)

def page\_not\_found(e):

return render\_template('not\_found.html'), 404

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

1. Создайте HTML страницы для каждого пункта меню и страницы 404. Например, index.html:

<!DOCTYPE html>

<html>

<head>

<title>Главная страница</title>

</head>

<body>

<h1>Добро пожаловать на сайт о путешествиях!</h1>

</body>

</html>

1. Запустите приложение, откройте браузер и перейдите по адресу [http://127.0.0.1:5000/.](http://127.0.0.1:5000/) Вы должны увидеть главную страницу сайта о путешествиях. Переходите по другим пунктам меню для просмотра соответствующих страниц.
2. Если ввести неправильный URL, то должна отображаться страница 404.

Таким образом, вы создадите сайт о путешествиях в Flask с четырьмя пунктами меню и обработкой ошибки 404.