

Lab 2: Flask Fundamentals

January 2023

In the practical section of this lab you will follow a tutorial on how to perform routing and URL handling, request and response objects, and templates and rendering views:

- Step 1: Routing and URL Handling

In the app.py file, import the Flask module and create a new Flask application object:

```
from flask import Flask
app = Flask(__name__)
```

Create a route using the @app.route decorator, and define a function to handle the request:

```
@app.route('/')
def index():
    return 'Hello World!'
```

To pass variables to the route, use angle brackets in the route and include a variable in the function:

```
@app.route('/users/<username>')
def show_user(username):
    return f'User: {username}'
```

To handle different types of requests, use the methods parameter in the @app.route decorator:

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        # handle login logic
    else:
        # show login form
```

- Step 2: Request and Response Objects

To access the request data, use the request object from the flask module and **replace** the previous login with the following:

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']

        if username == 'test' and password == 'test':
            return 'Login successful'
        else:
            return 'Invalid username or password'
```

To send a response to the client, use the `make_response` function and the response object:

```
from flask import make_response

@app.route('/')
def index():
    response = make_response('Hello World!')
    response.headers['Content-Type'] = 'text/plain'
    return response
```

- Step 3: Templates and Rendering Views

Create a new folder called `templates` in your application's root directory. Create a new file in the `templates` folder, for example `index.html`. Use Jinja2 syntax to create a template for your view:

```
<!DOCTYPE html>
<html>
<head>
    <title>My Application</title>
</head>
<body>
    <h1>Hello {{ name }}!</h1>
</body>
</html>
```

In the `app.py` file, use the `render_template` function to render the template and pass variables to it: Copy code from `flask import render_template`

```
@app.route('/')
def index():
    name = 'John'
    return render_template('index.html', name=name)
```

-
- Step 4: Adding the form template Create a new file in the templates folder called form.html .

```
<!DOCTYPE html>
<html>
<form method="post">
  <p><input type="text" name="username">
  <p><input type="password" name="password">
  <p><input type="submit" value="login">
</form>
</html>
```

You can now add the `/login` at the end of the website url to access the login page.

You should now have a basic understanding of how to do routing and URL handling, request and response objects, and templates and rendering views.

In case of any issues, you can refer to the official documentation of each tool for more detailed instructions and troubleshooting tips.