# Web Applications with FlaskRESTful

# Individual Project Assignment

This is the Individual Project Assignment for the **Web Applications with Flask - June 2022 @SoftUni**.

These requirements and evaluation are **only for the practical exam** (theoretical exam is not included in this assignment)

## REST Project (Flask REST project)

### General requirements

Your project **must** have all this functionality to pass the final examination.

* The application must be implemented using **FlaskRESTful Framework**
* The application must have at least **8 endpoints (up to 8 points)**
* The application must have **authentication and authorization functionality (15 points)**
  + The application must have **public part** (A part of the website, which is accessible by everyone – un/authenticated users and admins)
  + The application must have **private** part (accessible only by authenticated user or authenticated admins)
* **The application should have CRUD at least to a one resource a.k.a GET, POST, PUT and DELETE endpoint (not restricted to be for the same role) (10 points)**
* **The application should be structured, using MVC pattern or similar (different directories/packages for managers, services, routes/resources) (20 points)**
* **The input and output data should be validated and parsed against schemas (could be Marshmallow or similar) (15 points)**
  + **The application should have in general at least 2 custom and unlimited pre-built validators used in the schemas**
* **The application should follow the principle of class-based views and good OOP practices (10 points)**
* **Usage of ORM (flask\_sqlalchemy/sqlalchemy or similar) (15 points)**
* **The code should be formatted against pep8 standard (you can use black) (5 points)**
* **All imports should be in the correct order (you can use ctr+alt+o in PyCharm for automatic reordering) (2 points)**
* Use a **source control system** by choice **– GitHub is preferred**. (**10 points**)
* The application should use relational database for persistent storage (**20 points**)
* All third-party libraries/packages should be listed in requirements.txt file in the root folder of the project with their versions
* **For database credentials or other secret keys and data you should use environment variables which are not committed in the repo (or hardcoded in the code) (10 points)**
* **Gitignore file is mandatory to exclude all sensitive data, caches and etc. (5 points)**
* **At least one page of good described ReadMe file (should include how to install the dependencies, what are the endpoints, which are protected, what they return and what are the conditions to access them, the description of the project itself, future functionality) (5 points)**
* **At least one migration (up to 10 points – each migration is 2 points)**
* **Tests (30 points)**
  + **At least 5 meaningful commits**
  + **At least 5 tests of the most crucial feature of the app including mocking if needed**
  + **5 integration tests (api tests – from the request to this endpoint to the response)**
  + **Factories**
* **The application should be integrated with some 3th service of your choice (could be AWS S3 or AWS Simple email service, or could be a payment provider of your choice) (25 points)**

### Bonuses

* Write **tests** for at least **60% coverage** on your **business logic**
* Deployment
* CI or CD (with GitHub actions or Jenkins)
* Documentation/ Swagger
* Front end application (with framework like Angular, React VueJS or only vanilla JS)
* Different patterns with meaningful usage
* If the application is a creative app (something that helps you automate daily tasks, or we will be used by real users – your friends or family), something that solves an actual problem and has a potential to grow and be used
* Any other popular library like pandas, GraphQL and etc. with meaningful usage in the code

# Project Public Defense

Each student will have to deliver a **public defense** of its work in front of a trainer jury. Students will have **20-25 minutes**, which must be allocated as follows:

* **Demonstrate** how the application works (very shortly).
* **Prepared postman collection with working endpoints, exported for usage**
* Show the **source code** and explain how it works.
* Answer the jury's **questions**
* **Debug one endpoint live (chosen from the jury) and explain how each step is working**

Please be **strict in timing**! On the 10th minute, your presentation ends. The remaining time will be for Question/Answers session.

Open the project assets **beforehand** to save time.

Load the postman collection to save time.

Be **well prepared** for presenting the maximum of your work within the time given. It is highly recommended that you practice the presentation at home with a stopwatch, to be sure that you will fit in the time provided.

**Answering Questions**

* Answer questions from all Python's SoftUni program and potential functionality outside the scope of the project

**Bonuses – up to 15%**

## Submission Deadline

**Regular Exam – 28 August 2022**

* You **must** submit your project before 23:59 on the **25-Aug-2022**
* A presentation schedule will be available on the **26-Aug-2022** and will include only the projects that were **submitted beforehand**. Non-submitted projects will **NOT** be evaluated.

**Retake Exam – 4** **September 2022**

* You **must** submit your project before 23:59 on the **1-Sept-2022**
* A presentation schedule will be available on the **2-Sept-2022** and will include only the projects that were **submitted beforehand**. Non-submitted projects will **NOT** be evaluated.