Flask REST API

Secure Programming project

Overview of the program

- REST API implemented in Python with the Flask-RESTful framework
  - User can register with /users/register
  - User can save something called *slugs* with /secrets/<slug>
  - User can also retrieve the slugs by their ID or delete them by their ID with /secrets/<id>
- The API has no real use. It only represents a simple REST API
  - But it showcases everything a REST API does
    - HTTP methods GET, POST, DELETE
    - Uses JSON data
    - Works with an SQL database
    - Has some basic validation
    - HTTP Basic Auth

Using the application

There is no frontend, the API can be accessed with Postman or cURL.

```
HTTP POST User Registration
$ curl localhost:5000/users/register -X POST -H 'Content-Type: application/json' -d '{"username": "testi", "password": "averystrongandlongp…"}'
{"username": "testi"}

HTTP POST New Slug
$ curl localhost:5000/secrets/viesti -X POST -H 'Content-Type: application/json' -u testi:averystrongandlongpassword
{"id": "dfb180ad-30a6-40f8-b569-2695f937b258", "slug": "viesti", "user": "testi"}

HTTP POST Get Slug by ID
$ curl localhost:5000/secrets/dfb180ad-30a6-40f8-b569-2695f937b258 -H 'Content-Type: application/json' -u testi:averystrongandlongpassword
{"id": "dfb180ad-30a6-40f8-b569-2695f937b258", "slug": "viesti", "user": "testi"}
```

Security aspects in short

- HTTP Basic Auth (strong password authentication)
- Database ORM with SQLAlchemy (no straight SQL queries)
- UUID4 so that IDs cannot be guessed
- SHA-256 with a salt for password hashing
- Password strength requirements

Read me

More on the program can be read in its README.md.

It has the same testing examples as these slides as well as instructions for running the program. There you'll also find some security issues addressed.

https://github.com/sjaks/sepro-restful-flask