

# Самостоятельная работа №2.

---

## Простейший аккаунтинг.

Во многих веб-приложениях существуют набор стандартных операций, которые если и изменяются, то совсем незначительно. Одним из таких примеров является так называемый **AUP** или (**Authentication User Pipeline**). Суть его заключается в следующей связке:

- `/register -> /login`
- `/login -> /profile`
- `/profile -> /logout`
- `/profile -> /password/change -> /password/change/done -> /profile`

## Задание

Реализовать простейший пользовательский функционал для регистрации/аутентификации/обновления информации/смены пароля аккаунта.

## Модель

Задана модель пользователя(уже пользовались ею не раз):

```
class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(20), unique=True, nullable=False)
    email = db.Column(db.String(120), unique=True, nullable=False)
    image_file = db.Column(db.String(20), nullable=False, default='default.png')
    password_hash = db.Column(db.String(128), nullable=False)
```

## Функционал

- `/register` - страница формы регистрации. В форме регистрации запрашивается `username`, `email`, `password1`, `password2`. В случае если форма не валидна - оставляем на странице `/register` и выдаем `flash` уведомление `Invalid credentials. Try again`. В случае если форма валидна, создаем пользователя в бд, и перенаправляем его на страницу `/login`.
- `/login` - страница формы логина. В форме запрашивается для ввода 2 поля :`email`, `password`. В случае, если форма не валидна или такого пользователя не существует - оставляем на `/login` и выдаем `flash` уведомление `Invalid credentials. Try again`.
- `/profile` - страница профиля пользователя (запрос закрыт `@login_required`). Выводится текущая информация про пользователя (`username`, `email`), а также картинка профиля. На этой же странице располагается форма для обновления информации о пользователе. В ней находятся поля `username`, `email`, `picture`. Разрешенными форматами изображений считаем `.jpg`, `.jpeg`, `.svg`, `.png`. В случае валидности формы - редирект на `/profile` и `flash` уведомление `Profile`

updated successfully, в противном случае - редирект на `/profile` и `flash` уведомление `Error while profile updating`. Также на странице присутствует ссылка на `/password/change`.

- `/logout` - запрос с подтверждением - то есть при переходе на данную страницу отображается текст `Are you sure you want to logout?` и кнопка `LogOut` подтверждающая действие. После нажатия кнопки - разлогиниваем пользователя и перенаправляем на `/login`.
- `/password/change` - запрос закрыт `@login_required` декоратором. Данный запрос служит для изменения пароля пользователя. Содержит на себе форму с полями `Old password`, `New Password`, `New Password (Again)`. Форма валидна, если `Old Password` совпадает с текущим паролем пользователя, `New Password` и `New Password (Again)` не пусты и совпадают между собой. В случае валидности формы - редирект на `/password/change/done`. В противном случае - оставляем на `/password/change`.
- `/password/change/done` - запрос закрыт `@login_required` декоратором. Данный запрос выводит текст `Congrats. Your password was changed.`, а также имеется ссылка, отправляющая пользователя на `/profile`.

Реализовав данный функционал у вас на руках будет базовый аккаунт, которым можно будет пользоваться долгое время.

## Решение

Ваше решение должно содержать в себе весь исходный код проекта, `.env` файл, `.db` файл бд, `pipfile` и `pipfile.lock`. Решение загрузить в `github` репозиторий и ссылку на него прислать преподавателю в чат.