Домашнее задание №3

## Задача 1.

Вы принимаете участие в разработке подсистемы для управления пользователями. Каждый **Пользователь** имеет следующие атрибуты:

* **Идентификатор**. Тип данных – int. Уникальный, обязательный.
* **Email**. Адрес электронной почты пользователя. Используется для отправки почты и как логин пользователя в системе. Уникальный, обязательный.
* **Хеш пароля** (password hash). Произвольная строка. Используется для сверки введённого при входе в систему пароля. Обязательный.
* **Имя пользователя.** Произвольная строка. Обязательный.
* **Флаг «активен».** Тип данных – bool (bit). Обязательный.

Вам необходимо создать консольную утилиту, выполняющую следующие действия:

1. При старте утилита создаёт файл базы данных SQLite с именем **users.db**, а затем создаёт в этой базе таблицу **Users** с необходимой структурой (см. атрибуты Пользователя).
2. После создания базы и таблицы утилита вносит в таблицу **Users** данные пользователей из текстового файла **persons.txt**. В файле построчно перечислены пользователи – в каждой строке указаны Имя и Фамилия пользователя через пробел. Например,

Alexey Volosevich

John Doe

Mary Robinson

1. Правила заполнения таблицы **Users**:

* Идентификатор генерируется автоматически.
* Email формируется по шаблону [*FirstnameLastname*@coolcompany.com](mailto:FirstnameLastname@coolcompany.com).
* Хеш пароля вычисляется по Email при помощи функции GenerateFromPassword() из пакета bcrypt.
* Флаг «активен» устанавливается в true (1).

1. Если при разборе строки из файла **persons.txt** обнаруживаются некорректные данные, такая строка записывается в файл **errors.txt** вместе со своим порядковым номером.
2. После заполнения таблицы **Users** утилита выполняет контрольное чтение всех данных из этой таблицы и выводит их на консоль.

## Задача 2.

В вашем распоряжении находится файл базы данных SQLite с именем **users.db** (см. Задачу 1). Разработайте простейшее веб-приложение, позволяющее выполнить следующие REST-запросы (ответ с данными пересылается клиенту в виде JSON):

* GET /users – получить информацию (кроме хеша пароля) о всех пользователях.
* GET /users/id – получить инфо (кроме хеша пароля) о пользователе с Id = id.
* DELETE /users/id – удалить пользователя с Id = id.

## Задача 3.

В предыдущем домашнем задании вы создавали консольное приложения для управления списком задач – **TODO App**. **Дополните** (!) этот программный проект веб-приложением, отображающим список задач в виде таблицы. Пусть ваше веб-приложение работает, используя протокол HTTPS.