

**Проект № 26**

**Let's Learn Python**

**Направление: Софтуерни приложения**

**Автор:**

**Пламен Лъчезаров Трендафилов**

**ЕГН: 0641216485**

**Адрес: гр.София ж.к. Лагера бл.44**

**Телефон: 0876067114**

**E-mail: p.trendafilov@abv.bg**

**Клас: 9**

**Ръководител:**

**Екатерина Димитрова Мицева**

**Телефон:**

**E-mail:**

**Длъжност: Учител по информатика и ИТ**

**1. Цели и целева група**

**1.1. Цели**

Приложението е с цел учителите да могат да правят свои собствени задачи и да ги дават на учениците си. В системата може да се правят, редактират и изтриват задачи, да се дават на учениците и да се тестват автоматично.

**1.2. Целева група**

Учители, училища, ученици и заинтересовани, които обучават, учат програмния език Python.

**2. Основни етапи**

**2.1. Проучване на съществуващи решения по темата**

**2.2. Съставяне на план за реализация на проекта**

- 2.3. Определяне на функциите, които приложението ще изпълнява
- 2.4. Планиране на структура и архитектура на проекта
- 2.5. Проучване на необходимите технологии и начини за осъществяване на проекта
- 2.6. Създаване и разработване на приложението
  - 2.6.1. Направа на отделните елементи
  - 2.6.2. Свързване на елементите
  - 2.6.3. Създаване на отделните страници
  - 2.6.4. Създаване на базата данни
  - 2.6.5. Свързване към базата данни
  - 2.6.6. Създаване на функционалност
  - 2.6.7. Редактиране на направените страници
  - 2.6.8. Редактиране на направените функционалности
- 2.7. Тестване и проверка за грешки
- 2.8. Публикуване в интернет
- 2.9. Въвеждане в експлоатация
- 2.10. Популяризиране

### **3. Ниво на сложност**

За да създам приложението трябвах познания за създаване на динамичен уеб сайт и с заявки от сайта към сървъра. Технологиите, с които разработвах проекта са HTML, CSS, JavaScript, JQuery, PHP и MySQL. Най-трудното нещо в проекта беше да компилирам правилно кода от конзолата към сървъра и да върна отговор. В началото бях решил сървърната част на тази функционалност да бъде на Python, но повечето пъти кодът не успяваше да се компилира правилно, затова смених сървърната част и на тази функция да бъде на PHP.

### **4. Логическо и функционално описание на проекта**

#### **4.1. Логическо описание**

- Приложението има три функционалности като една от тях изисква връзка с локален сървър, който сървър да компилира кода.
- *Вход:* В него се намира функционалността за направа и влизане в акаунт, всеки един потребител може да си направи акаунт.

- *Създаване на задачи:* Друга функционалност, след като сме си влезли в създаден акаунт е да правим задачи, да видим какво сме направили, да редактираме нашите задачите и да ги изтриваме.
- *Всяка отделна вече създадена задача, която се визуализира на началната страница:* Третата функционалност е да решим вече създадена задача, като се праща код, като решение на задачата, след това се проверява дали решението е вярно и се връща отговор, дали задачата е вярна или грешна.

#### 4.2. Функционално описание:

- 4.2.1. Създаване на акаунти, с които да се създават, редактират и изтриват задачи
- 4.2.2. Регистрация и вход в акаунтите, чрез които да запамятаваме кои са задачите създадени от нас.
- 4.2.3. Пращане на решения на задачите във вид на код и проверка дали кодът изпълнява дадената функционалност.

#### 4.3. Нефункционални изисквания:

- Приложението е независимо
- Приложението е с потребителски интерфейс.
- Сайтът е с адаптивен дизайн за използване от различни телефони, таблети, лаптопи, компютри и др.
- Приложението работи с всички браузъри
- Потребителският интерфейс интуитивен
- Приложението е винаги достъпно
- Приложението проверява кодът и подсказва къде може да е грешката в него
- Приложението проверява данните за регистрация и ако има грешка, се подсказва на потребителя да я поправи

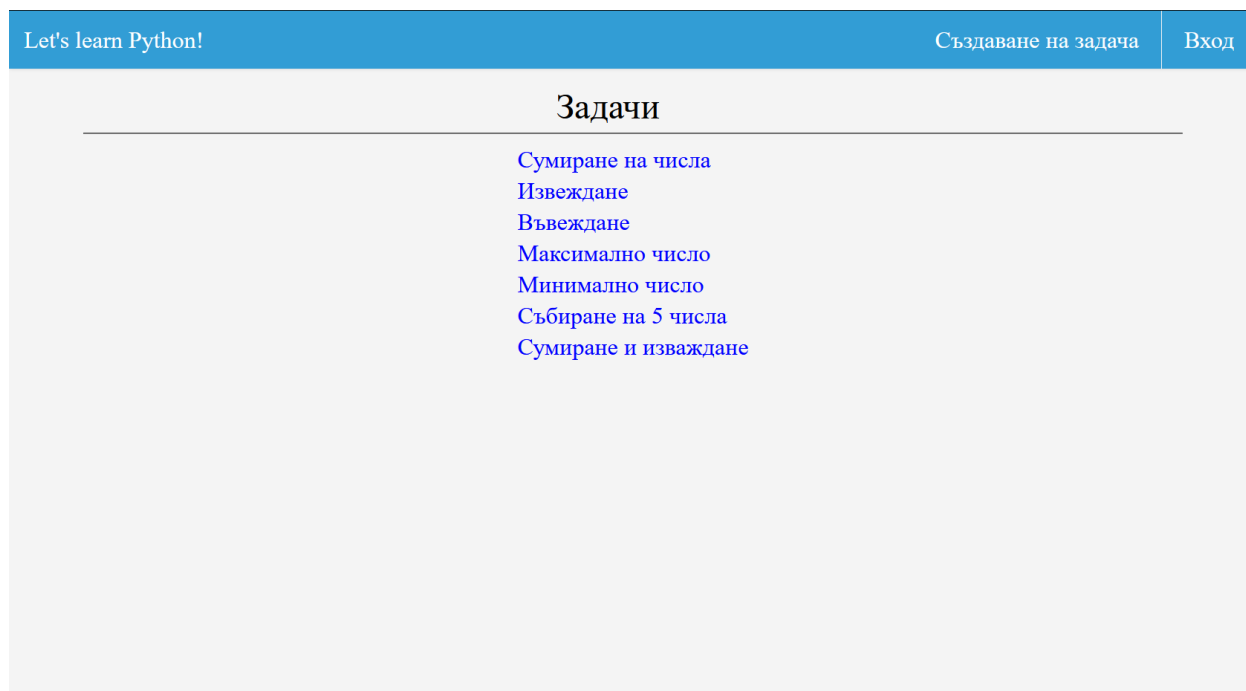
### 5. Реализация

- Клиентската част на приложението е направена на HTML, CSS и JavaScript
- Основните функционалности са написани на JavaScript и PHP
- Базата данни е реализирана чрез phpMyAdmin, а кодът е написан на MySQL

## 6. Описание

Приложението се намира на адрес <https://smart-eel-41.locat.it/>

Описание на интерфейса:



В дясно се намира менюто, което има следната функция:

- Потребителят да се регистрира или да си влезе в акаунта.
- Когато потребителят си е влезнал в акаунта може да създаде задача

## Let's learn Python!

### Създай задача

Заглавие:

Условие на задачата:

Входни данни:

Изходни данни:

Показки:

- След като сме създали задача може да я редактираме или изтрием от менюто, което е заменило **Вход**, то е същото, което сме задали за username в регистрацията

Let's learn Python!

Създаване на задача

UsernameOlymp

## Задачи

- Сумиране на числа
- Извеждане
- Въвеждане
- Максимално число
- Минимално число
- Събиране на 5 числа
- Сумиране и извеждане

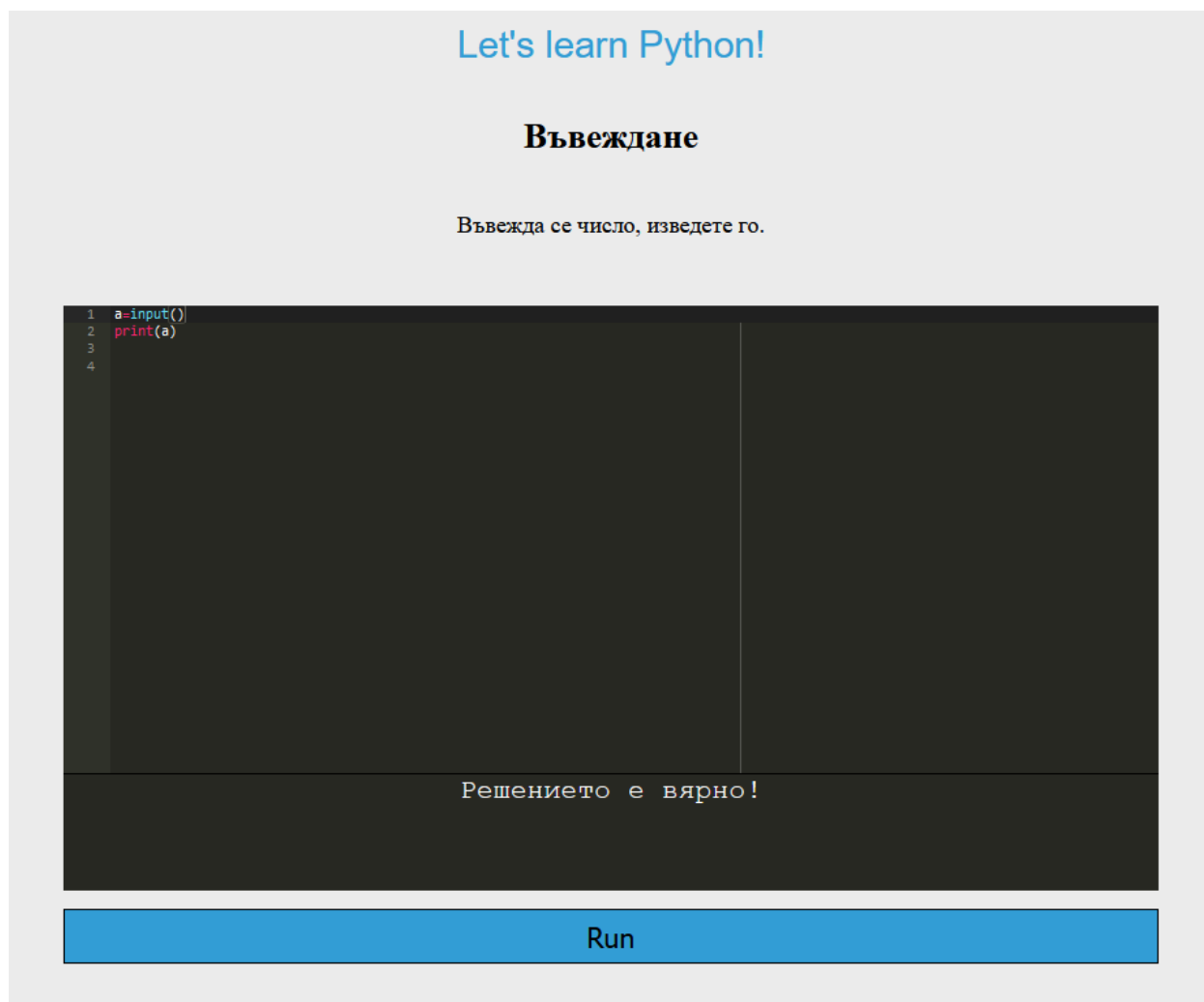
46.249.73.155/www/index.php?token=ctue

Следа натискането на username ни (В случая UsernameOlymp), се визуализира друга страница с функциите да изтрием, редактираме задача или да си излезем от профила.

Мои задачи					
Заглавие	Условие на задачата	Входни данни	Изходни данни	Подсказки	Действие
UserOlymp	UserOlymp	UserOlymp	UserOlymp	UserOlymp	<a href="#">Edit</a> <a href="#">Delete</a>
UserOlymp1	UserOlymp1	UserOlymp1	UserOlymp1	UserOlymp1	<a href="#">Edit</a> <a href="#">Delete</a>

[Излез от профила](#)

- Вече създадени задачи (Сумиране на числа, извеждане, въвеждане и др.) при натискане на някоя от тях се визуализира заглавието на задачата и нейното условие, а под условието конзолата, където потребителят въвежда кода си и го проверява.



## 7. Заключение

- Учителите разполагат с платформа за създаване на задачи, на която могат да публикуват задачите си и след това ученик или потребител може да я реши.
- Решаването е направено интуитивно все едно работят с интегрирана среда и кодът им се проверява за секунди дали прави, заложената от учителя функционалност.