**1.Тема: Училищен органайзер за ученици – приложение за организиране на училищни дейности на ученици**

**2.Автори:**

**Имена: Мартин Ивелинов Попов**

Егн: 0144238068

Адрес: гр. Добрич ул. Йордан Йовков №6

Телефон: +35988-872-5372

Имейл: [m\_p1@abv.bg](mailto:m_p1@abv.bg)

Училище: Езикова Гимназия „Гео Милев“

Клас: XI – в

**Имена: Траян Ангелов Райковски**

Егн: 0241250968

Адрес: гр. Добрич ул. Страцин №4

Телефон: +35988-680-8842

Имейл: [traqn02@gmail.com](mailto:traqn02@gmail.com)

Училище: Езикова Гимназия „Гео Милев“

Клас: XI – в

**3. Ръководител:**

Имена : Мария Кирилова Георгиева

GSM : 0895355091

Имейл: [eg.m.kirilova@gmail.com](mailto:eg.m.kirilova@gmail.com)

Старши учител

**4. Резюме:**

**4.1 Цели:**

Целта на нашия проект е да улесни начина на организиране на училищната програма и дейности от страна на учениците. Досега всеки ученик трябва да запише учебната си програма на лист или в бележника и всеки път, когато иска да провери кой е следващият час трябва да погледне какво е записал на съответния хартиен носител. Нашето Уеб-приложение предоставя възможността това да става просредством електронен вариант на училищен органайзер. Освен седмична програма, той включва и списък с контролни, въвеждани от потребителя, възможност за въвеждане на оценки и тяхното складиране плюс автоматично показване на следващ час, заедно с оценки за седмицата и контролни или други събития, в период от две седмици от този ден.

Идеята ни е да създадем платформа в която учениците могат да си организират бързо и лесно програмата и други дейности, свързани пряко с училище. Вместо да се записва програмата и всеки път да се търси къде е, по-лесно е да се отвори телефонът и с едно натискане да се отвори, заедно с появата на следващия час.

**4.2 Oсновни етапи в реализирането на проекта:**

Разгледахме първоначалната идея заедно с нашия ръководител г-жа Кирилова. После ние я отработихме и доразвихме, добавяйки множество функции.

Миналата година изучихме доста технологии, като основните на които се спряхме са Angular 2+ за front-end, а за back-end - Golang. Така тази година знаехме какво ще ползваме за front- и back-end и не губихме ценно време в учене и усвояване. Новото, което разучихме, е AWS Lambda. Това е безсървърен сървър, който няма физически сървър.

Ние си разделихме работата. Първоначално един от нас направи една част от front-end-a, а другият разучаваше как да направи back-end-a, понеже тепърва се запознавахме с тази нова технология, която е на по-малко от година. Това предполага много проблеми с всичко и работата с нея наподобява бета тестинг. След това се сменихме, за да може всеки от нас да знае какво се случва с всяка част от проекта. Понеже сме в един клас имахме възможност да разговаряме за проблемите и да си помагаме.

**4.3 Ниво на сложност на проекта:**

Тази година решихме да надграждаме старите знания, като добавим нещо ново към тях. Вече имахме сравнителен опит с Angular и затова проблемите възникваха в областта на back-end-a и по-точно - самият сървър на който се качва. AWS Lambda има много оскъдна документация специално за Golang, понеже той е сравнително нов език и затова няма много информация за интеграцията му с тази платформа. Поради тази причина около два месеца минаха в тестове, разработка и качване на сървърната част на проекта.

**4.4 Логическо и функционално приложение:**

**Front-end:**

Идеята в Angular е да има много модули, като всеки модул си има различно предназначение. Всеки модул се състои от папка с .html , .ts, .css файл.

**Модули:**

Core – компонент, който е активен винаги освен ако не е сме в page-not-found

Home – начална страница

Login - логин страница

Register – регистриране на нов потребител

Not-found – ако потребителят въведе страница, която не съществува, той бива препратен към този компонент

Profile – профилът на потребителя, където може да управлява профила си

В папка shared се съдържат файлове който се използват от много компоненти и които са посредниците между back-end и front-end

//add modules

**Back-end:**

**//rewrite everything**

**4.5 Реализация:**

Още в началото знаехме какво ще ползваме за реализирането на проекта. Това са технологиите: Angular 2+, Golang, DynamoDB е базата данни, а AWS Lambda ни е сървърната платформа. С Angular се правят Single Page Applications (SPAs). Това значи че нашият back-end е REST API.

**//fix add service worker**

**4.6 Описание на приложението:**

**//fix desc**

Нашият проект е уеб-приложение. За да се използва сайтът, трябва да се направи профил. Ако потребителят иска да предлага услугите ще се наложи да даде допълнително информация, като адрес на офиса, магазина,име на бизнеса и други. Хората ще могат да си записват часове, като отворят даден бизнес и подадът заявката, която после бива приета или отхвърлена от собственика.

**4.7 Заключение:**

Имаме още идеи за подобряване на приложението, които възнамеряваме да реализираме в близкото бъдеще:

* да добавим групи, в зависимост от интереси(Например: по немски език), клас или други фактори
* потребители с различни привилегии(учители, директори)

Надяваме се, че нашият проект ще помогне на учениците да организират своята училищна програма и всякакви дейности, свързани пряко с училище. Ще се радваме, ако проектът ни да помогне на всеки ученик.

**//add more**

фр