[Наслов]

Содржина:

Што е целта на оваа дипломска работа?

Корисничка перспектива

* Регистрација
* Логирање
* Ажурирање на профил
* Прикачување на корисничка слика
* Безбедност
* Нотификации по е-пошта
* Студенти
* Компании
* Регионалност
* Меко бришење

Техничка перспектива

* Кои технологии се искористени?
* Каква е структурата на базата?
* Каква е структурата на кодот?
* Како е имплементирана автентикацијата?

Тестирање

* Какви тестови имаме?
* Како се имплементирани тестовите?

Понатамошен развој

* Што може да се подобри?
* Што може да се додаде?
* Каков бизнис модел може да се имплементира?

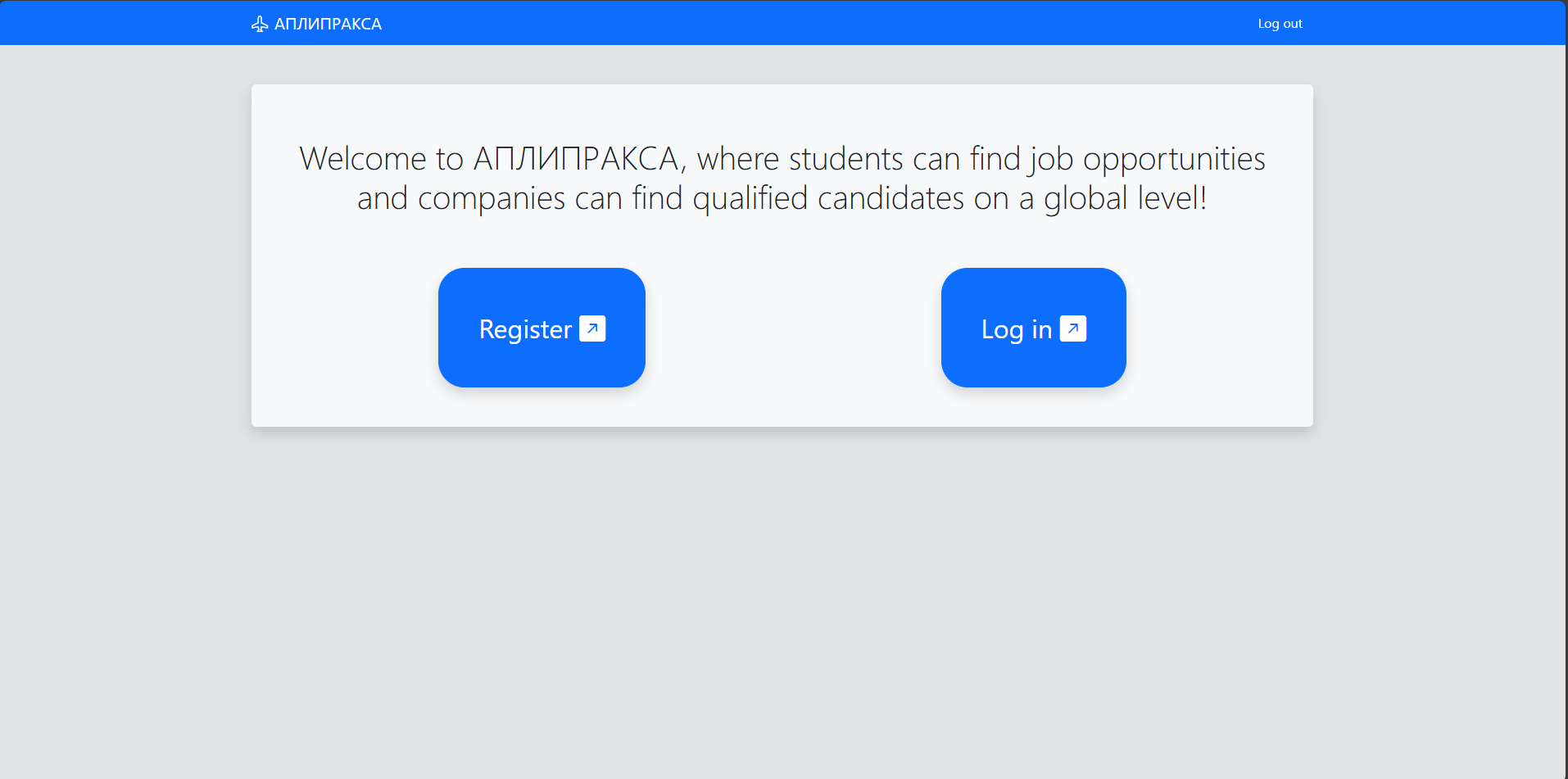
**Што е целта на оваа дипломска работа?**

Целта на оваа дипломска е да се изработи функционална веб апликација наменета за два типа на корисници – компании и студенти, која ќе овозможи нивно поврзување преку објавени понуди за пракса. Компаниите би можеле да објавуваат свои понуди за пракса, да разгледуваат и селектираат апликанти, односно студенти. А студентите би можеле да ги разгледуваат истите понуди и да аплицираат на нив. Веб апликацијата ќе биде изработена со помош на следните технологии: PostgreSQL, Python, Psycopg, FastAPI, Jinja Templates, Javascript, Bootstrap, HTML, Bash и Pytest.

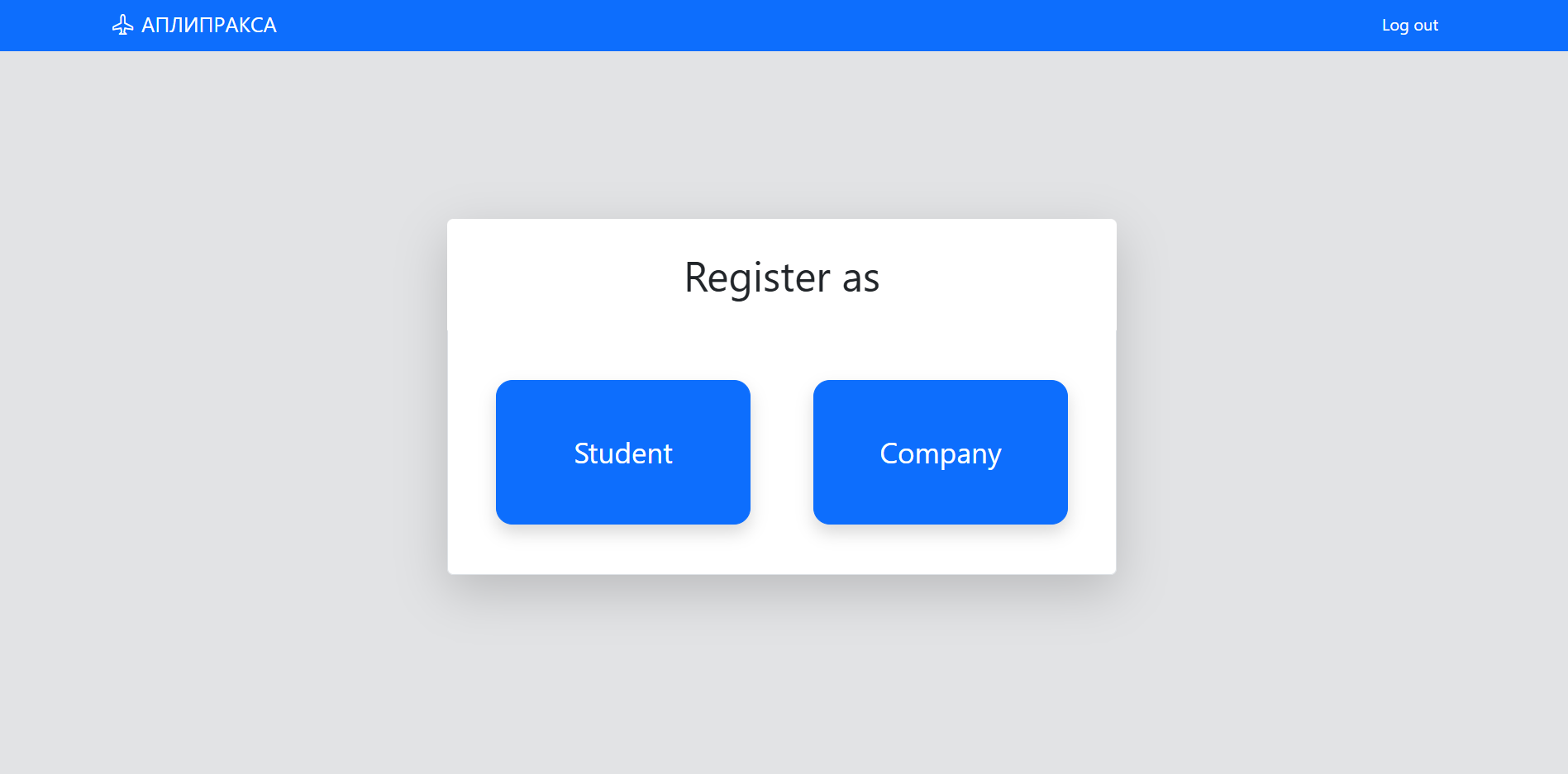
**ДЕЛ 1: КОРИСНИЧКА ПЕРСПЕКТИВА**

**Регистрација**

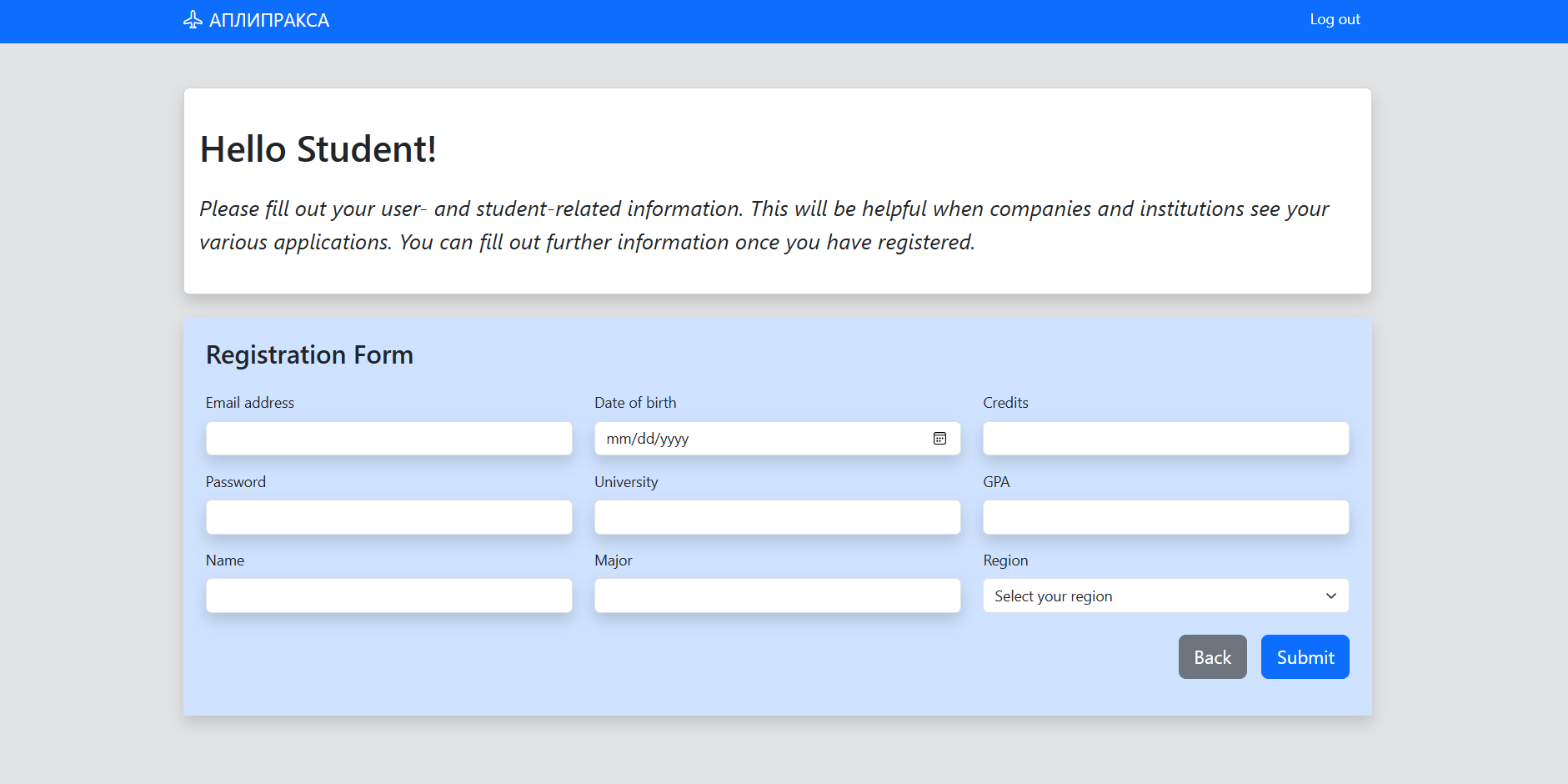
Кога ќе ја отвориме апликацијата веднаш сме насочени кон почетната страна. Тука понудени сме две опции – да се регистрираме со нов кориснички профил или пак да се најавиме доколку веќе имаме креирано свој профил.



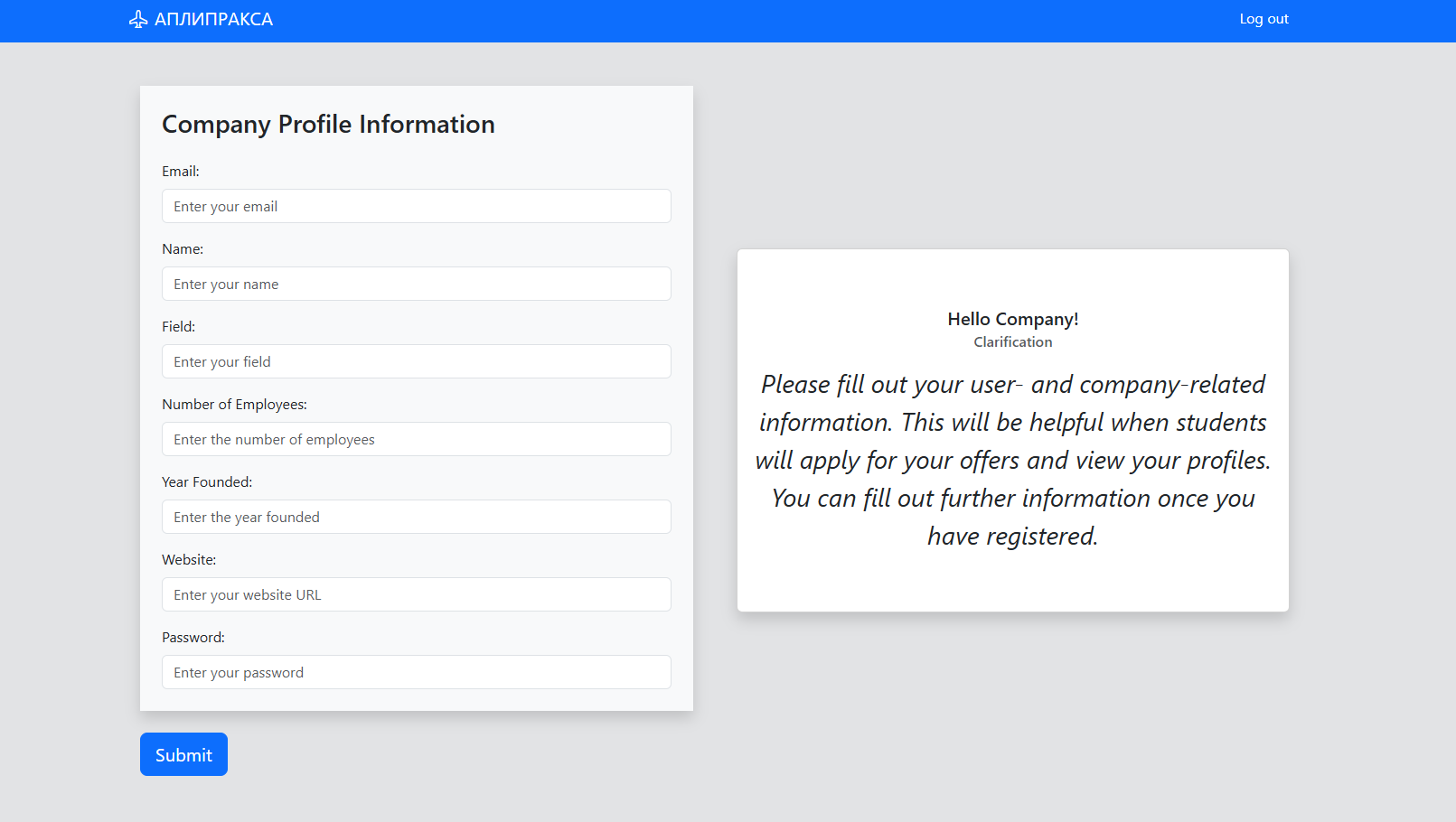
Доколку ја одбереме опцијата за регистрација, ќе бидеме пренасочени на наредната страна, каде дополнително може да одбереме каков тип на кориснички профил сакаме да одбереме, односно дали сакаме да се регистрираме како студент, кој ќе може да разгледува понуди за пракса, или пак како компанија, која ќе може да објавува понуди за пракса.



Доколку го одбереме корисничкиот тип – студент, ќе ни се покаже страна со форма за пополнување на нашите кориснички студентски информации: име, универзитет, смер, просек, број на кредити, е-пошта, лозинка, и сл.



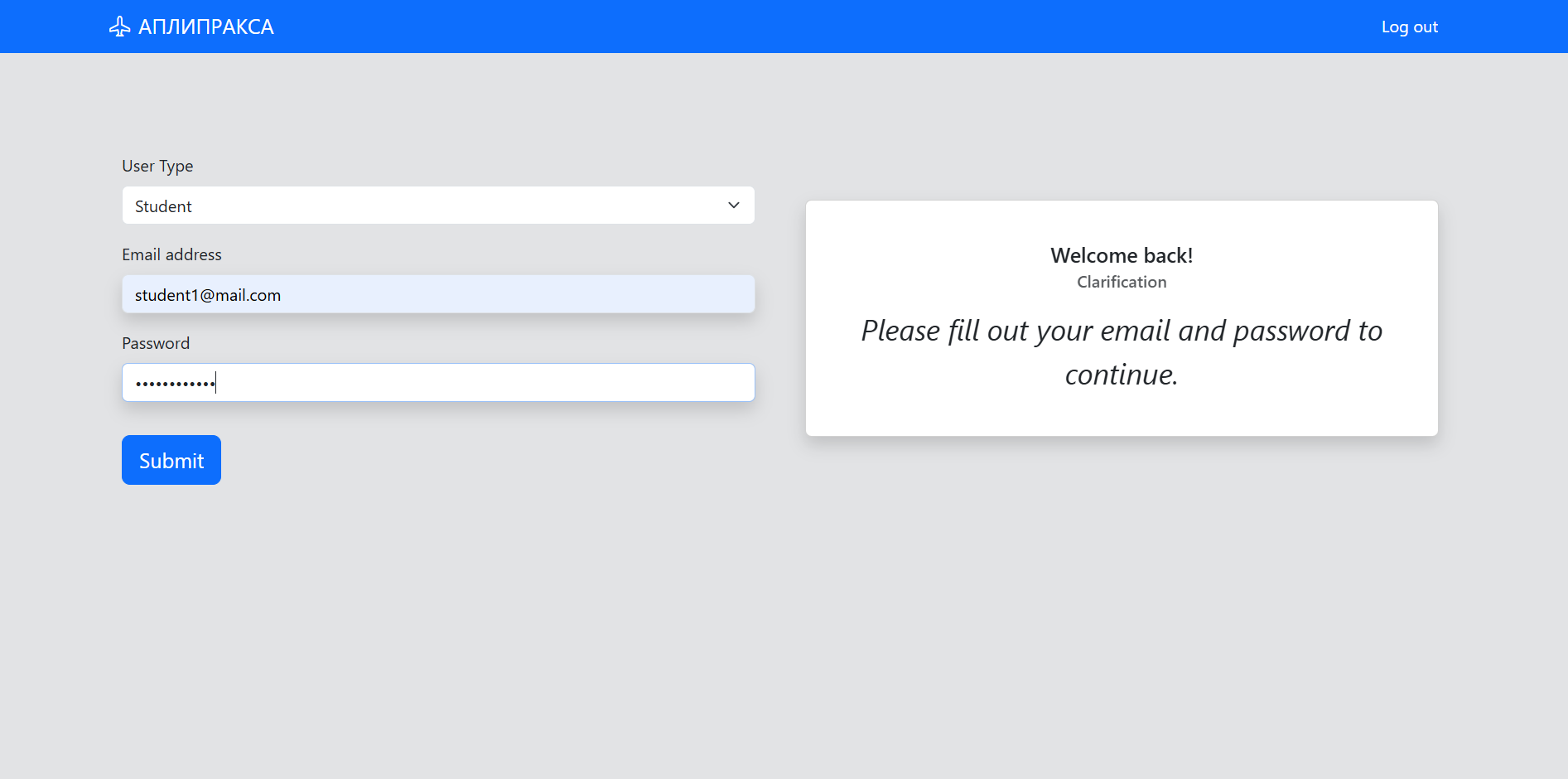
Доколку го одбереме другиот кориснички тип – компанија, ќе ни се покаже слична страна со форма за пополнување на корисничките информации на компанијата: име, е-пошта, лозинка, област на работење, број на вработени и сл.



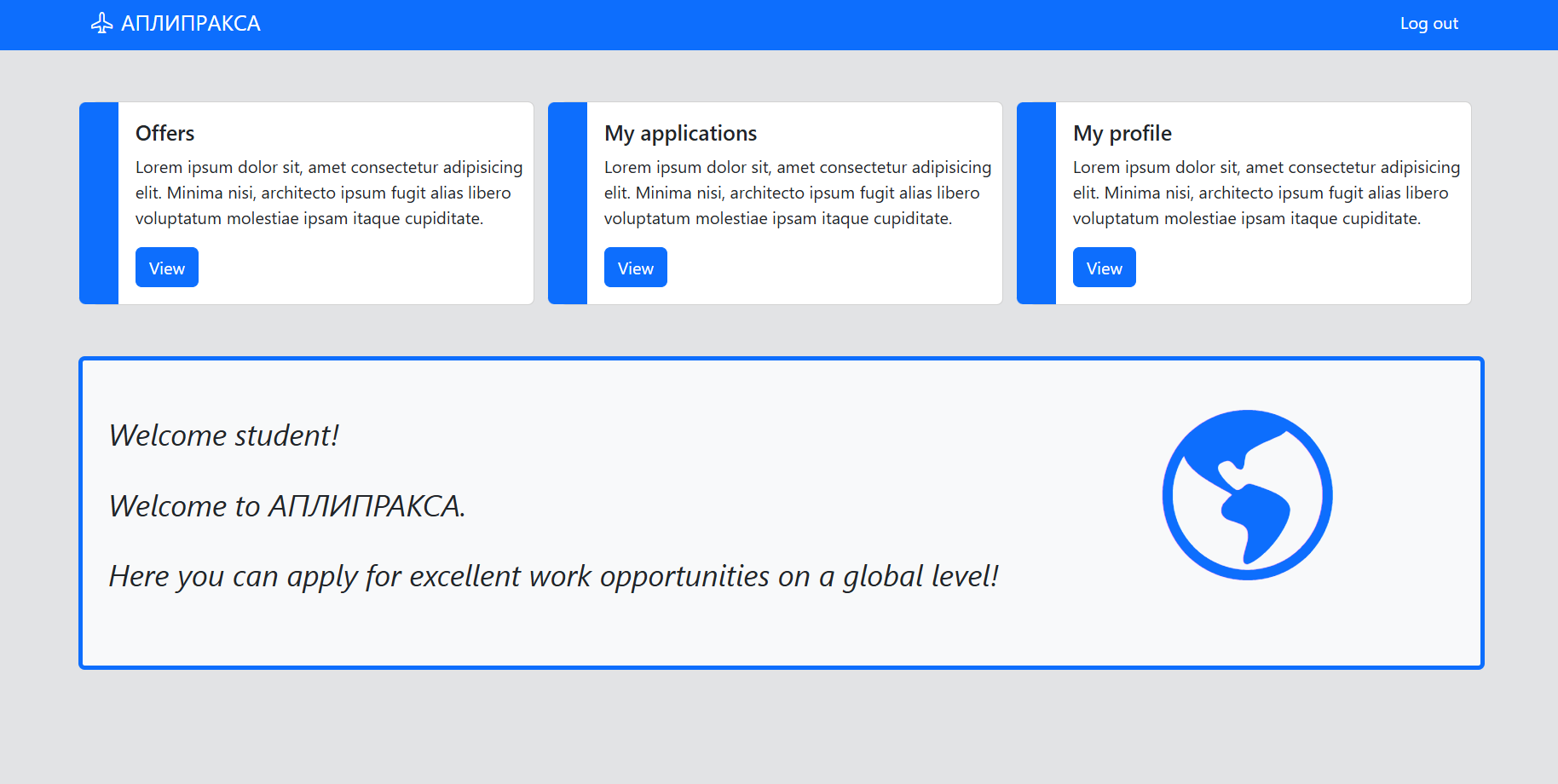
Дел од информациите кај двата типа на корисници се преклопуваат, но поголемиот дел се разликуваат. Сите овие информации ќе се прикажуваат и ќе можат да се ажурираат во понатамошниот дел од корисничките сценариа.

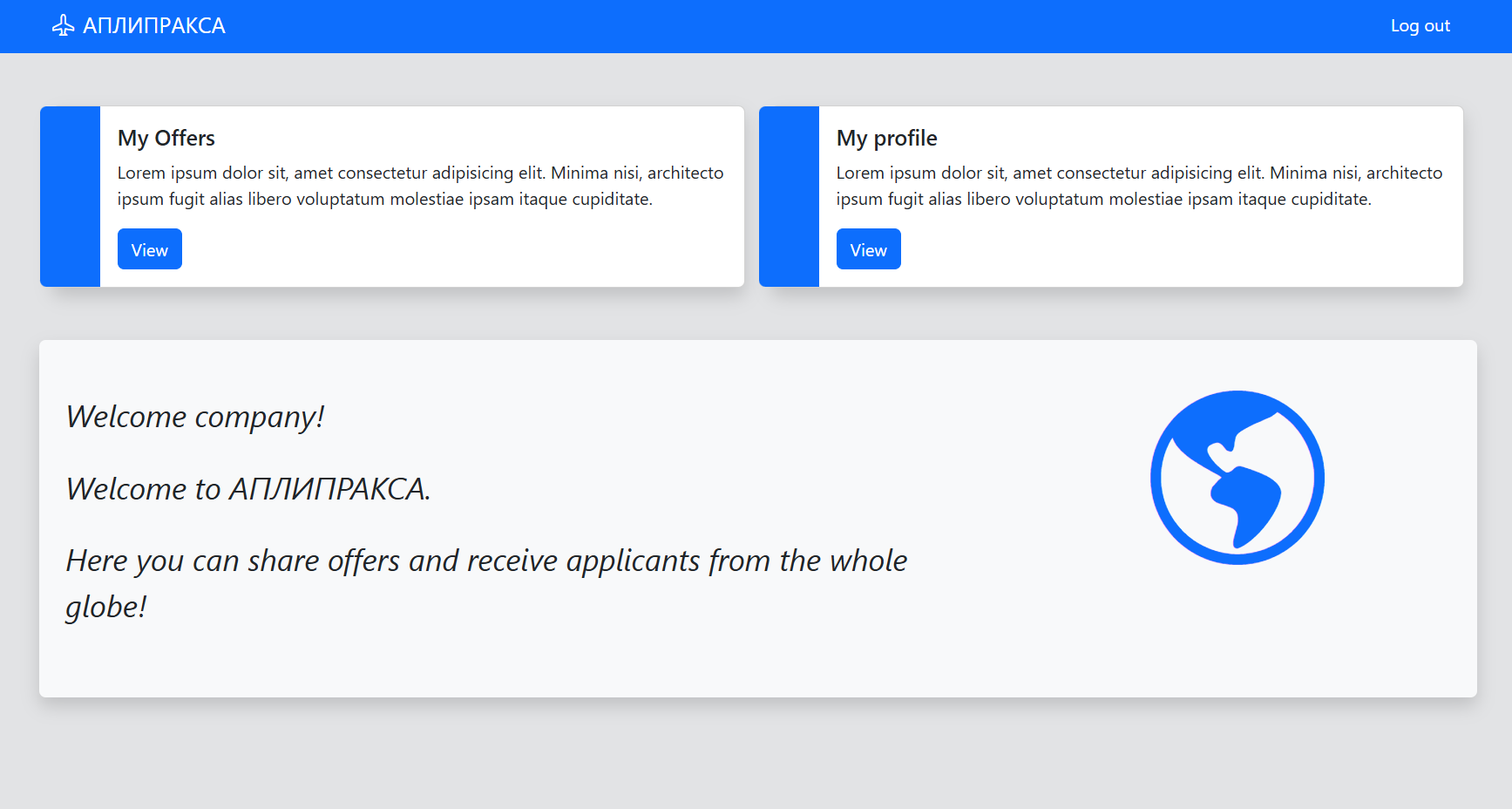
**Логирање**

Доколку студент или компанија веќе има креирано свој профил, кога повторно ќе ја отвори апликацијата, ќе може да се најави на системот со својата е-пошта и лозинка, и притоа селектира кој тип на корисник е – студент или компанија.



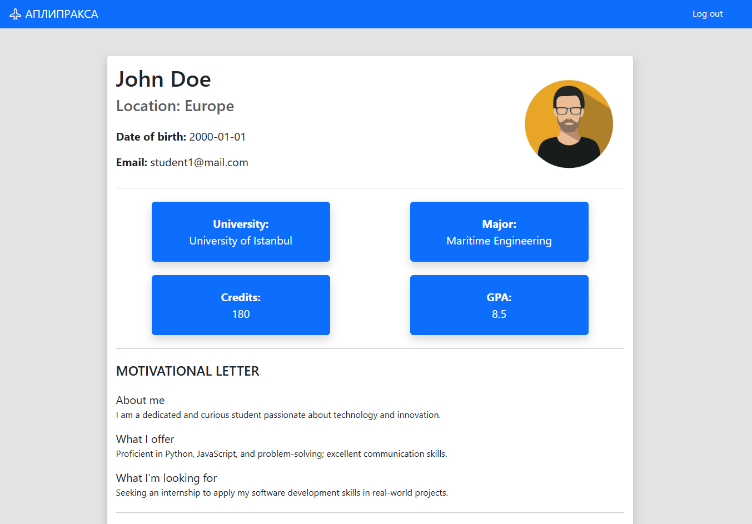
Откако ќе го изврши ова, корисникот ќе биде насочен кон своја почетна страна каде ќе може да ги користи останатите функционалности на апликацијата. Студентите и компаниите имаат слични, но различни почетни страни со различни опции за користење.

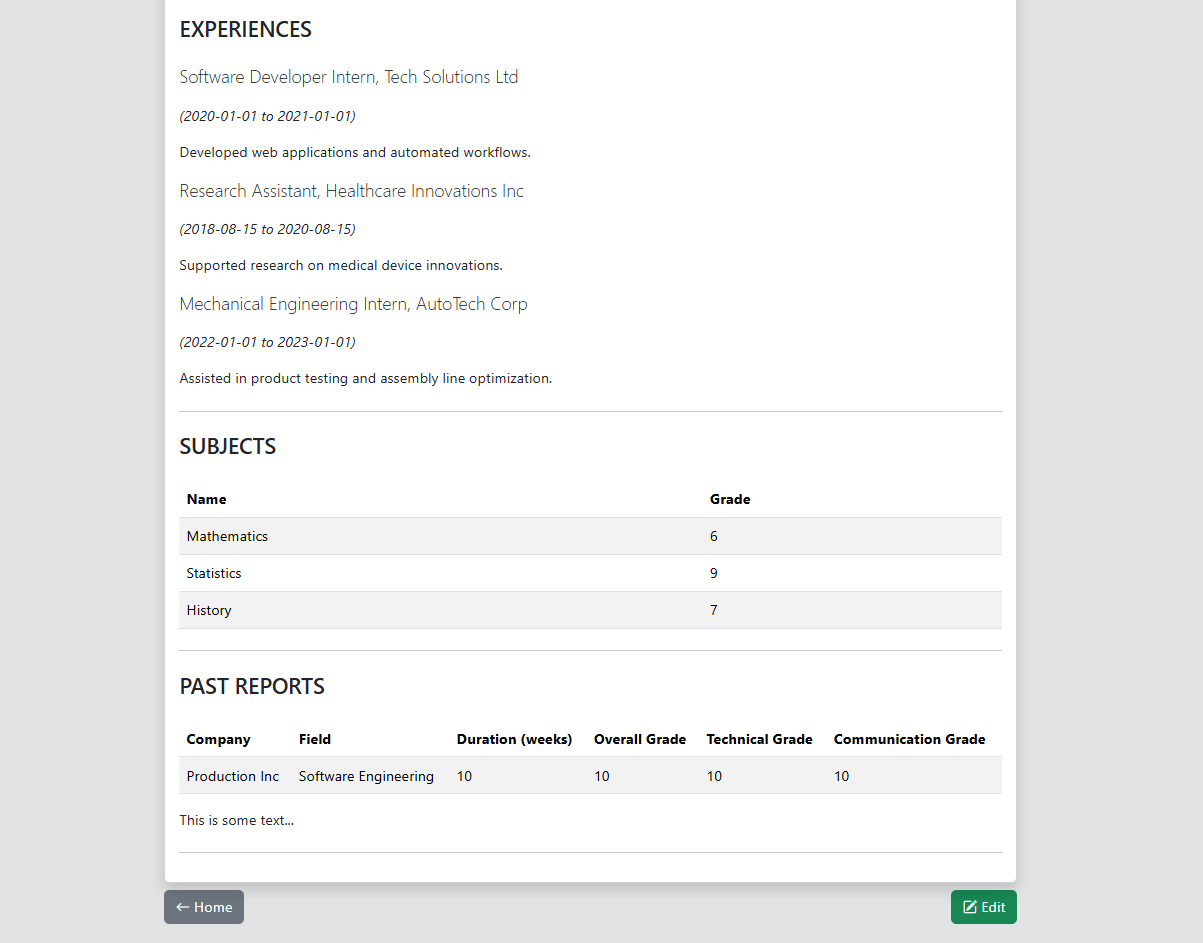


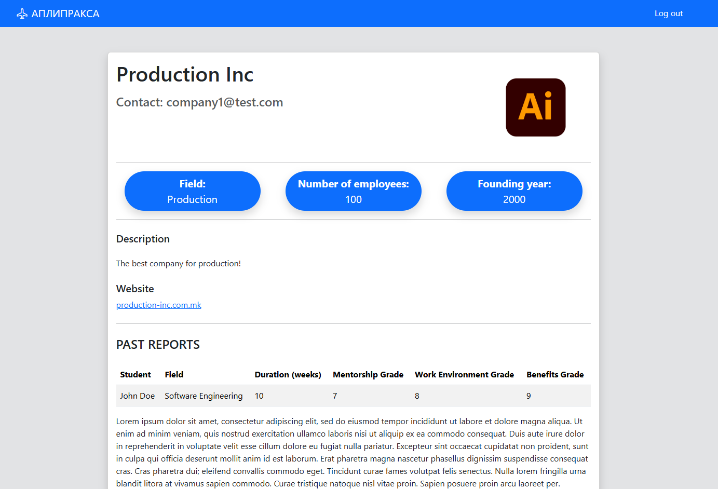


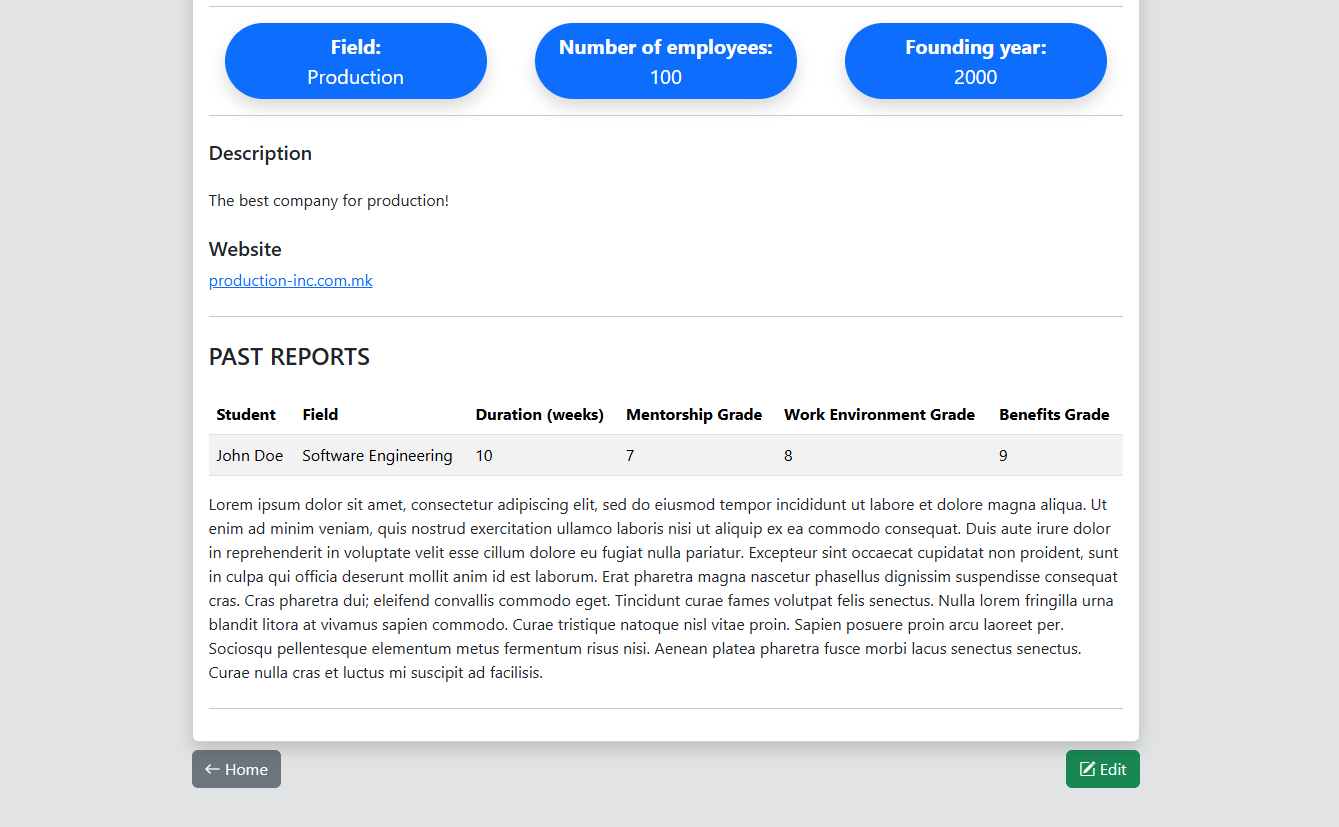
**Ажурирање на профил**

Двата типа на корисници можат да си го разгледаат својот профил на ист начин како што би се прикажувал на останатите корисници (дел од понатамошните функционалности). Можат потоа да кликнат на копче за ажурирање на својот профил.

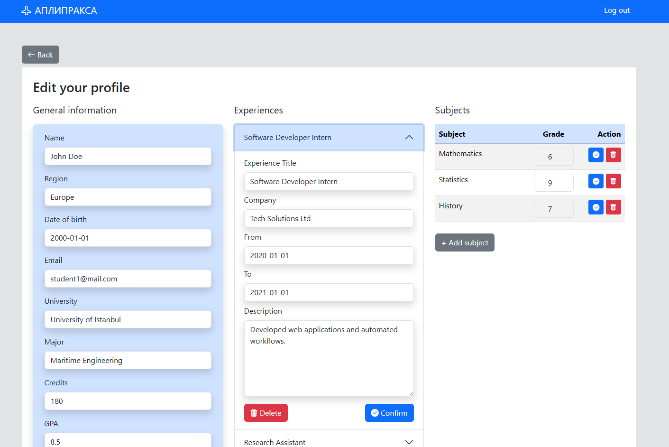


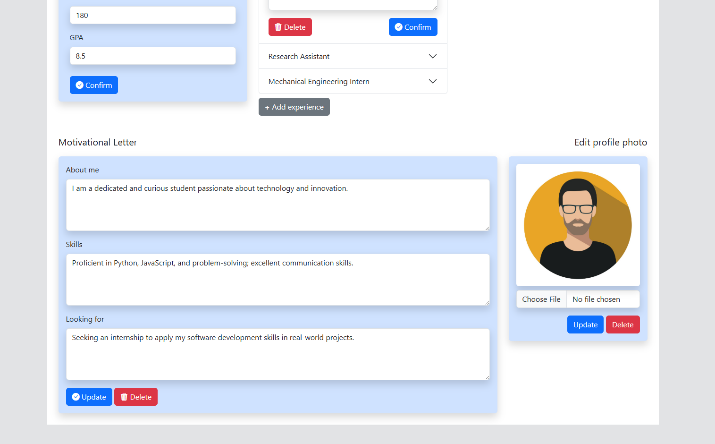


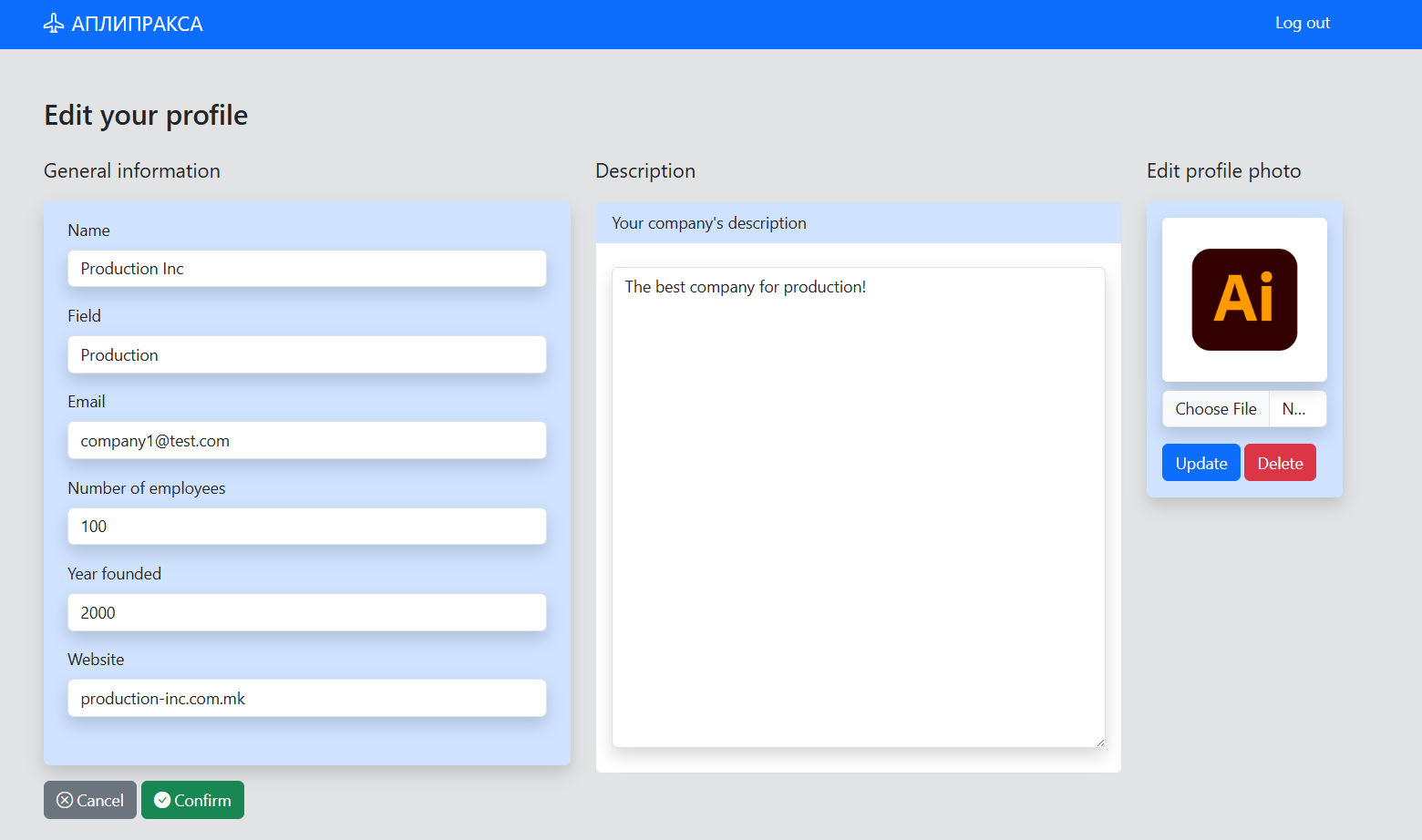




Кога ќе го кликнат тоа копче, ќе бидат пренасочени кон нова страна што ќе ги има истите информации за нивниот профил, но сега во рамки на форма за внесување на податоци. Ќе можат да си ги променат своите информации и да кликнат на копче за потврдување.







По ова, корисникот ќе биде автоматски вратен на страната за приказ на неговиот профил, со цел да има увид како би изгледала промената на останатите корисници.

Дополнително, студентите имаат можност да креираат, ажурираат и бришат други елементи од нивниот профил, односно нивниот профил опфаќа повеќе информации кои ќе бидат потребни во понатамошниот тек на користење на апликацијата – за ова ќе стане збор во наредните делови.

**Прикачување на корисничка слика**

Во рамки на функционалностите за ажурирање на информации за својот профил, корисниците исто така можат да си прикачат корисничка слика која ќе се прикажува на секој што ќе може да го отвори профилот. Доколку нема корисникот до сега прикачено слика, ќе се прикажува default слика со празна сива икона – ова со цел да се одржи визуелниот интегритет на апликацијата.

Ќе можат исто така корисниците да си ја променат сликата со прикачување на нова дадотека, или едноставно да си ја тргнат корисничката слика. Доколку си ја тргнат корисничката слика, автоматски ќе се појави оригиналната default слика со сива икона.

**Безбедност**

Од помалку видливиот, но можеби и од покритичниот аспект на апликацијата преземени се мерки за одржување на безбедноста на системот.

***Хеширање на лозинки***

Првата клучна мерка е изведена при самото регистрирање на корисниците. Корисникот при внес на неговите податоци – конкретно при внес на неговата е-пошта и лозинка – автоматски ќе бидат зачувани истите податоци во базата на апликацијата (за да може повторно да се најавува на системот). Но тоа што не го гледа корисникот е дека неговата лозинка не се зачувува во ист формат како што е внесена. Се извршува хеширање на лозинката и хешираната вредност се зачувува во нашата база. Пример:

***Информации внесени од корисникот:***

*Е-пошта:* [*student.user@mail.com*](mailto:student.user@mail.com)

*Лозинка: 123*

***Зачувани вредности во база:***

*Е-пошта:* [*student.user@mail.com*](mailto:student.user@mail.com)

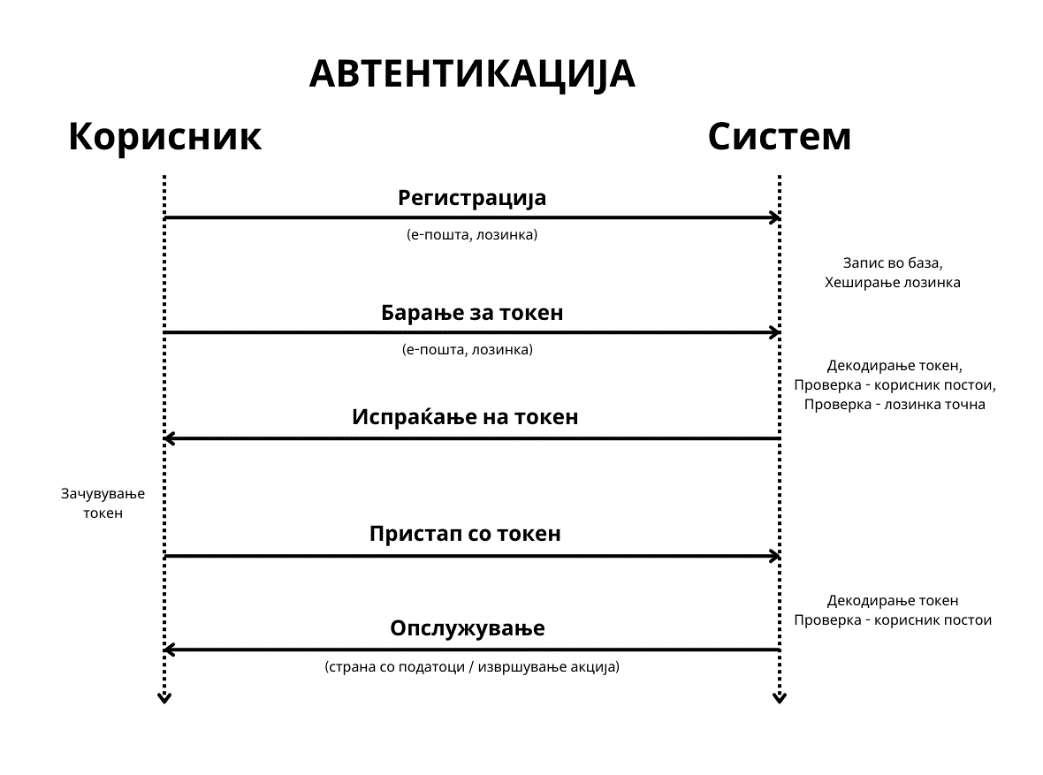
*Хеширана лозинка: $2b$12$zRR8lvkdcmiHjn.*

Зошто се прави ова? Ова е стандардна мерка што се употребува во секој кориснички систем со цел доколку некој напаѓач некако успее да добие пристап до информациите во базата, да не може да ја открие вистинската лозинка на корисникот (во примерот тоа е 123). Хеширањето ја постигнува оваа цел бидејќи е еднонасочна операција.

***Автентикација***

Втора мерка преземена за безбедност на системот е имплементирана веднаш по самото регистрирање корисникот или при неговото регуларно најавување, откако веќе има креирано профил. Оваа мерка е процесот на автентикација, односно потврдување на идентитетот на корисникот и дека тој е веќе дел од списокот на корисници кои се регистрирани на апликацијата.

Зошто е ова потребно? За речиси сите функционалности кои следат по регистрација или логирање, би сакале да овозможиме пристап до апликацијата само доколку некој некој е веќе најавен на системот, односно не би сакале било кој анонимен корисник да може да ги користи услугите на апликацијата.



Како го постигнуваме ова? При самото регистрирање или логирање на корисникот, приморан е да побара токен од апликацијата така што ќе ја приложи својата е-пошта и лозинка. Доколку корисникот приложи е-пошта и лозинка кои не се дел од записите во базата, нема да добие никаков токен. Потоа за сите понатамошни пристапи до апликацијата, ќе биде побарано од корисникот да го приложи добиениот токен. Доколку не се приложи токенот, ќе биде оневозможен пристап до соодветната страна / услуга на апликацијата.

***Авторизација***

Последната безбедносна мерка е авторизација, која е имплементирана во самиот процес на користење на апликацијата. Авторизација е слично како автентикација, но во овој чекор веќе сме провериле дали моменталниот корисник е валиден и постои во системот. Сега треба да се провери дали моменталниот корисник има дозвола да ја изврши посакуваната акција.

Зошто е ова потребно? Едноставно не сакаме сите корисници на системот да имаат дозвола за користење на сите можни услуги. На пример, не треба да дозволиме корисник А да може да ажурира податоци за профилот на корисник Б, и обратно.

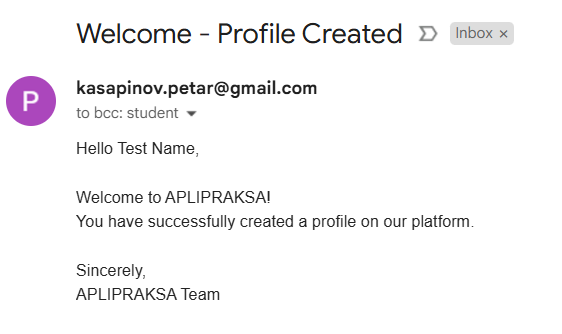
[ДИЈАГРАМ ЗА ИМПЛЕМЕНТИРАНА ЛОГИКА ЗА АВТОРИЗАЦИЈА]

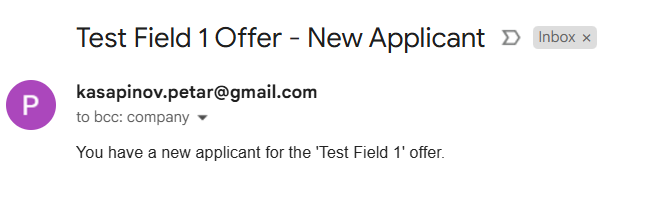
Како го постигнуваме ова? Повторно користиме токени за потврдување на идентитетот на корисникот. Но сега, не само што ќе провериме дали приложениот токен соодветствува со регистриран корисник, туку мора да имаме и дополнителна логика за проверување дали тој корисник смее да изврши одредена акција. Не може да имаме генерализирана логика за овој чекор, бидејќи за секоја акција може да има сосем различно правило. На пример, за ажурирање на информации за профилот, треба да провериме дали тој профил припаѓа на најавениот корисник, но за некоја друга услуга – на пример, креирање на понуда за пракса, доволно е само да провериме дали корисникот е од тип компанија, а не од тип студент.

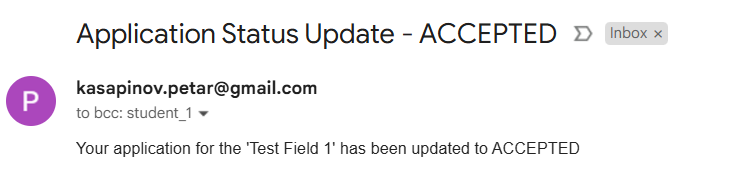
**Нотификации по е-пошта**

Покрај услугите за време на користење на самата апликација, имаме имплементирано и праќање нотификации по е-пошта. Нотификација ќе биде испратена до е-поштата која ќе биде внесена од корисникот при неговото прво регистрирање, и тоа при следните чекори:

* Креирање на профил – до студенти и компании
* Добивање на нов студент апликант за пракса – до компании
* Прифаќање на апликација за пракса – до студенти
* Одбивање на апликација за пракса – до студенти
* Откажување на прифатен студент – до компании
* Ресетирање на статус на апликација – до студенти







За секој различен чекор, ќе биде испратен различен темплејт на порака со динамички внесени податоци, како на пример: корисничко име, нов статус на апликацијата, и слично.

**Студенти**

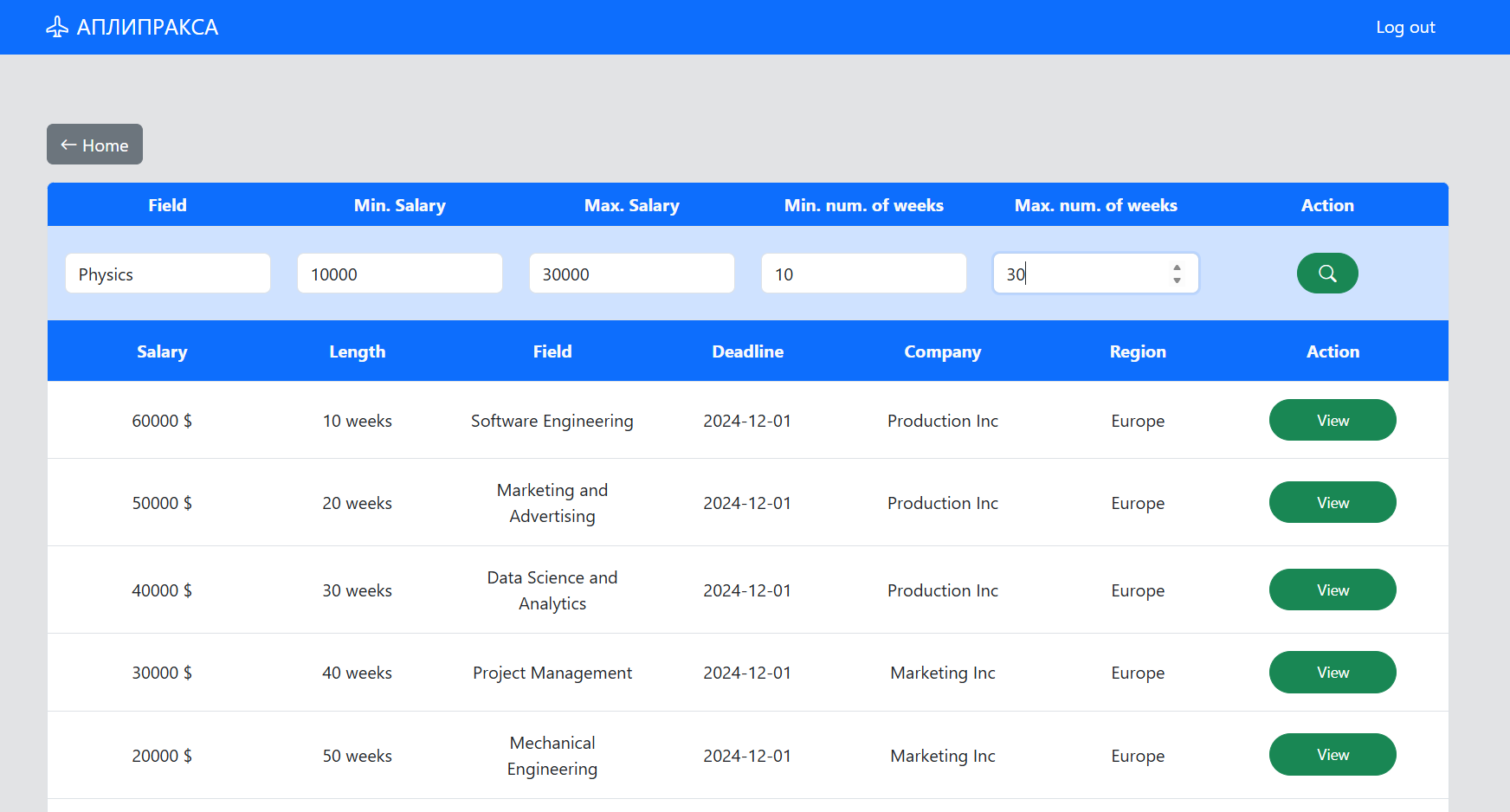
До сега ги претставивме функционалности кои генерално беа заеднички за корисниците студенти и корисниците компании. Но примарната цел на нашата апликација е да понуди услуги до различен тип на корисници, па со тоа и самите услуги ќе се разликуваат.

За студентите имаме одреден број на услуги кои се овозможени само за нив. Ги имаме следните:

* Разгледување на понуди
* Филтрирање на понуди
* Разгледување на понуда
* Разгледување профил на компанија
* Аплицирање на понуда
* Откажување на апликација
* Разгледување на свои апликации
* Креирање извештај за компанија

***Разгледување на понуди***

На нивната почетна страна студентите можат да видат три секции: Offers, My Applications, и My Profile. Доколку го одберат делот Offers, ќе им се појави список на сите понуди за пракса на кои можат да аплицираат. За секоја понуда ќе биде прикажана кратка содржина со клучните информации: понудена плата, времетраење, поле, рок за аплицирање, име на компанијата, регион на извршување на праксата. Исто така за секоја пракса има копче View кое ќе биде клучен дел за понатамошните функционалности.

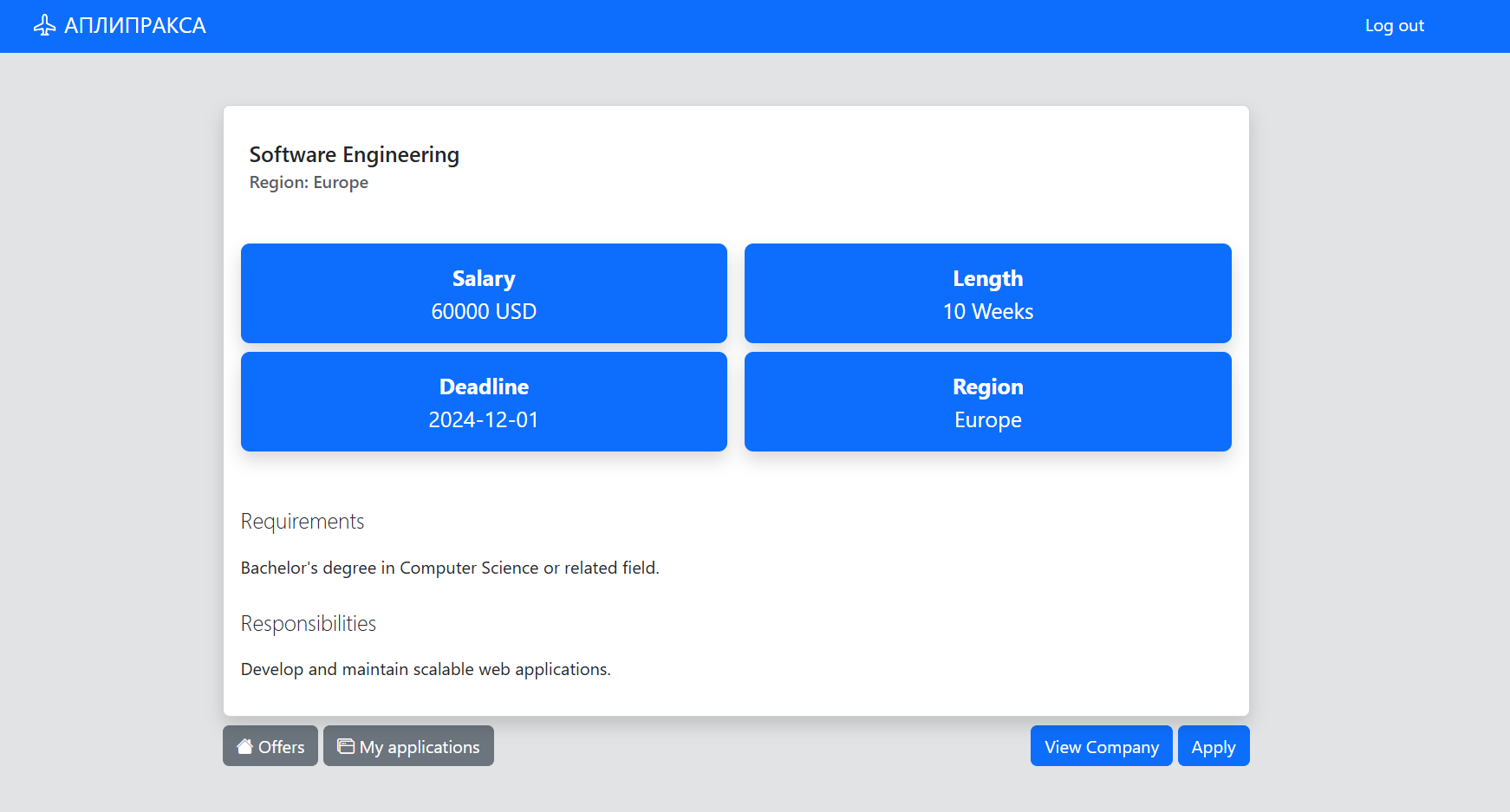


***Филтрирање на понуди***

Над самиот список на сите видливи понуди, студентот ќе има поле за дефинирање на параметри за филтрирање на понудите. Ова е потребно доколку има преголем број на можни понуди, односно за студентот да разгледува само понуди кои се релевантни за него. Може да филтрира според: поле на праксата, минимална плата, максимална плата, минимално времетраење, и максимално времетраење. Откако ќе се внесат сите параметри, студентот треба да кликне на иконата за пребарување, по што ќе се прикажат само соодветните понуди. Доколку сакаме да го ресетираме филтерот и да ги гледаме повторно сите можни понуди, потребно е да кликнеме на истата икона без да внесеме филтер-параметри.

***Разгледување на понуда***

Доколку студентот кликне на копчето View за одредена понуда, ќе биде пренасочен на друга страна каде ќе може да ги разгледа информациите за понудата во повеќе детали. Ќе бидат прикажани истите информации кои беа дел од крактата содржина: поле, регион, плата, времетраење, и рок. Дополнително ќе бидат прикажани и подолги текстуални секции за барањата и одговорностите кои се однесуваат на праксата.



***Разгледување на профил на компанија***

За секоја ваква отворена понуда, студентот ќе може да кликне на копче View Company за да ги разгледа подетално информациите за самата компанија која ја објавила понудата. Го пренасочуваме корисникот на друга страна, каде ќе бидат прикажани: името на компанијата, нивната е-пошта, поле на компанијата, бројот на вработени, година на основање, нивната корисничка слика, веб страна, и опис. Всушност ова е истиот приказ што компанијата може отвори за својот профил, само сега на студентот му е оневозможено да премине на страната за ажурирање на профилот.

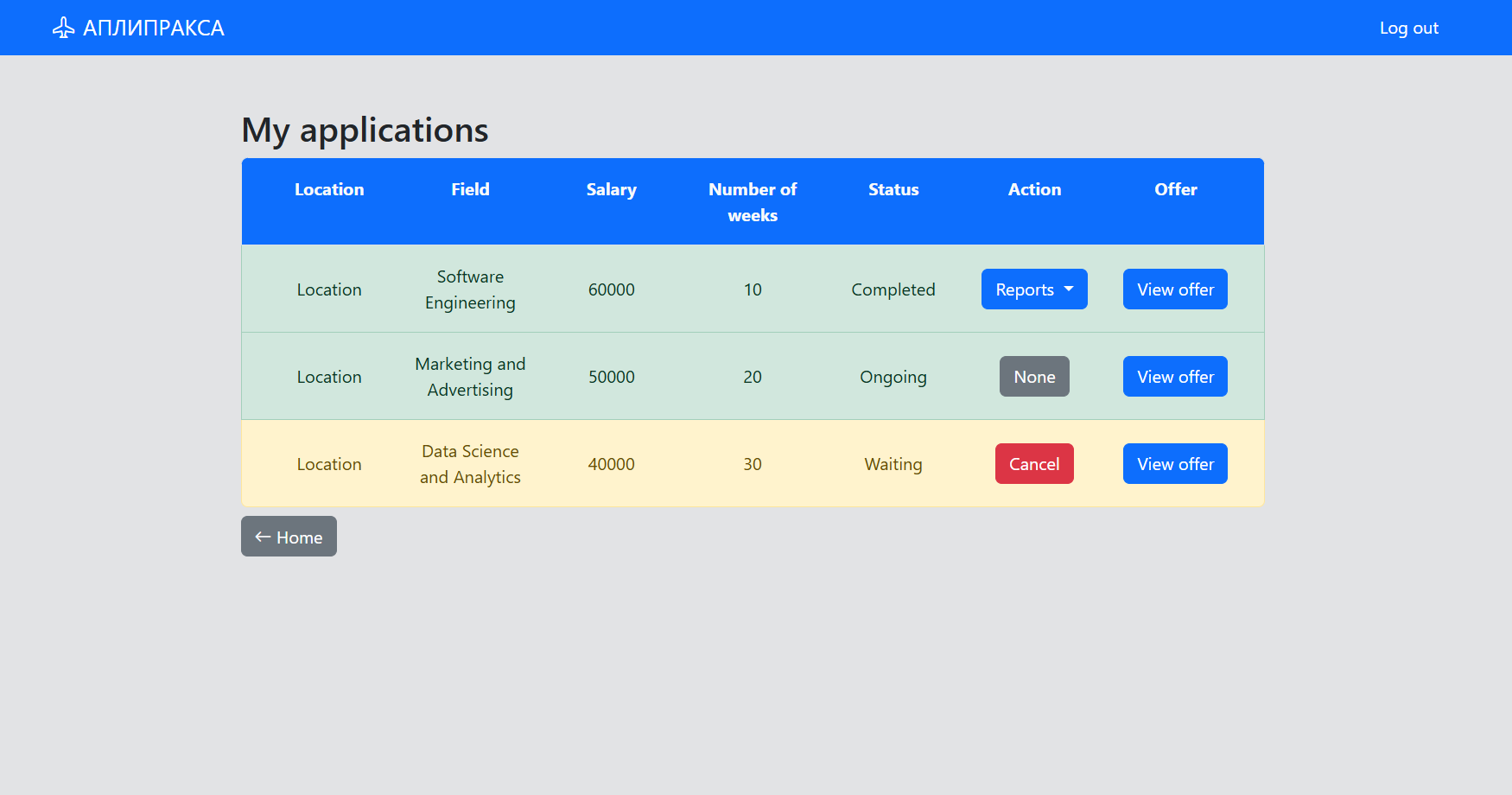
На оваа страна исто така има дел за приказ на поранешни извештаи креирани од студенти што извршиле пракса кај компанијата – за ова ќе стане збор подоцна.

***Аплицирање на понуда***

На претходната страна, каде беше прикажана целата понуда, има и копче Apply. Кога студентот ќе го кликне, ќе се отвори прозорче каде ќе биде прашан дали е сигурен дека сака да аплицира на понудата. Може да се откаже или пак да потврди. Доколку потврди, ќе биде автоматски пренасочен на друга страна каде ќе се прикажуваат сите негови апликации – подолу објаснето повеќе. Значи во тој момент студентот веќе аплицирал за понудата и неговата апликација ќе има статус Waiting и ќе биде обележана со жолта боја. Овој чекор ќе биде видлив и за компанијата, така што ќе може во својот дел да ја види истата апликација.

***Разгледување на свои апликации***

За поголема прегледност, секој студент може да си ги отвори сите негови досегашни апликации.



На неговата почетна страна има секција My Applications. Доколку ја отвори секцијата, ќе се покаже нова страна со список од сите понуди на кои аплицирал досега, заедно со кратка содржина за секоја понуда, боја на апликацијата која го означува статусот – во чекање (жолто), одбиен (црвено), прифатен, започнато, завршено (зелено). За секоја апликација може повторно да се отвори целосната содржина на понудата преку клик на копчето View offer. Исто така за секоја апликација има копче за извршување на одредена акција која ќе биде поврзана со моменталниот статус на апликацијата:

* Cancel – за апликации со статус Waiting или Accepted
* Reports – за апликации со статус Completed
* None – за апликации со Rejected и Ongoing

За ова ќе наведеме повеќе детали во наредните делови.

***Откажување на апликација***

Доколку студентот поради некоја причина се одлучи да ја откаже апликацијата, тој може тоа да го направи. На страната каде се наведени сите негови апликации има колона Action, каде се појавува копче за интеракција со понудата. Ќе има различно копче во зависност од моменталниот статус на апликацијата.

Доколку има статус Waiting или Accepted, ќе биде прикажано црвено копче Cancel. Со клик на ова копче, соодветната апликација се тргнува од списокот на студентот.

При откажување на апликацијата, сите останати студенти кои веќе биле одбиени за истата понуда, поради тоа што веќе првично бил прифатен моменталниот студент, сега ќе имаат обновен статус Waiting и ќе можат понатаму да бидат прифатени од страна на компанијата. Секако, моменталниот студент нема видливост на овие промени кај другите студенти.

За да се одржи историски интегритет на податоците, студентот не може да откаже апликација за која е веќе одбиен – статус Rejected, или пак која е веќе започната или завршена – статуси Ongoing и Completed.

***Креирање извештај за компанија***

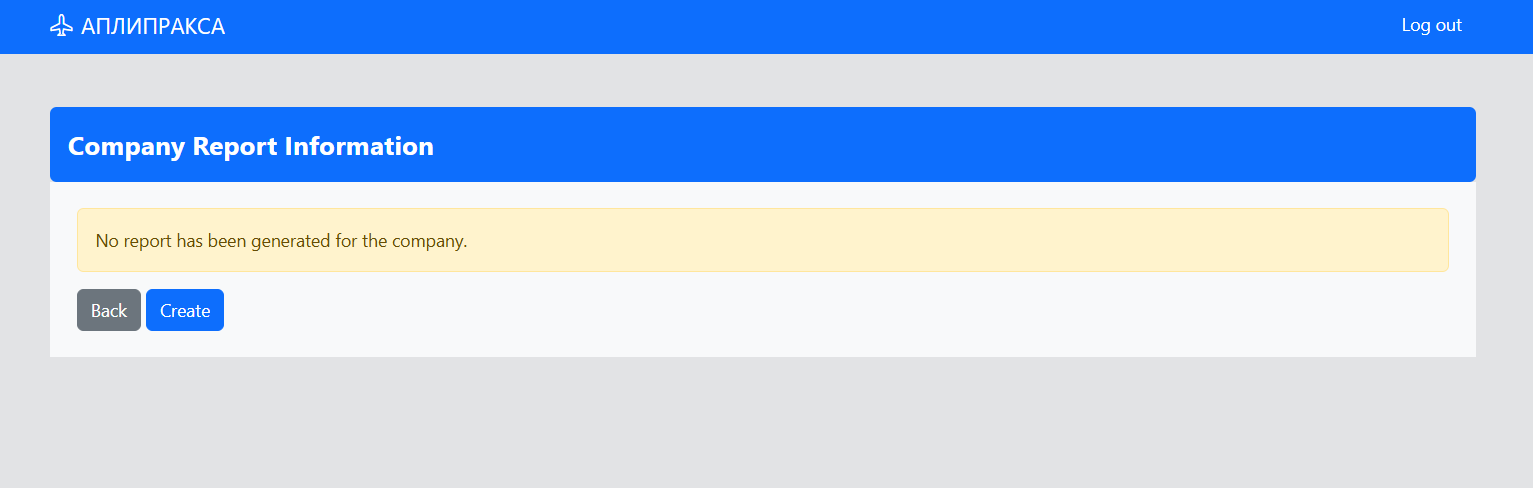
За да имаат сите понатамошни корисници на системот поголема транспарентност за квалитетот на компаниите кои нудат пракси, и за да можат студентите да можат да добиваат конструктивна критика за нивните извршени задачи, имплементиран е систем на креирање извештаи за извршени пракси.

Откако статусот на прифатена апликација ќе се ажурира од Accepted во Ongoing, па повторно во Completed, за студентот ќе се појави нова акција Reports на неговата страна за апликации. Со клик на ова копче ќе има можност да одбере:

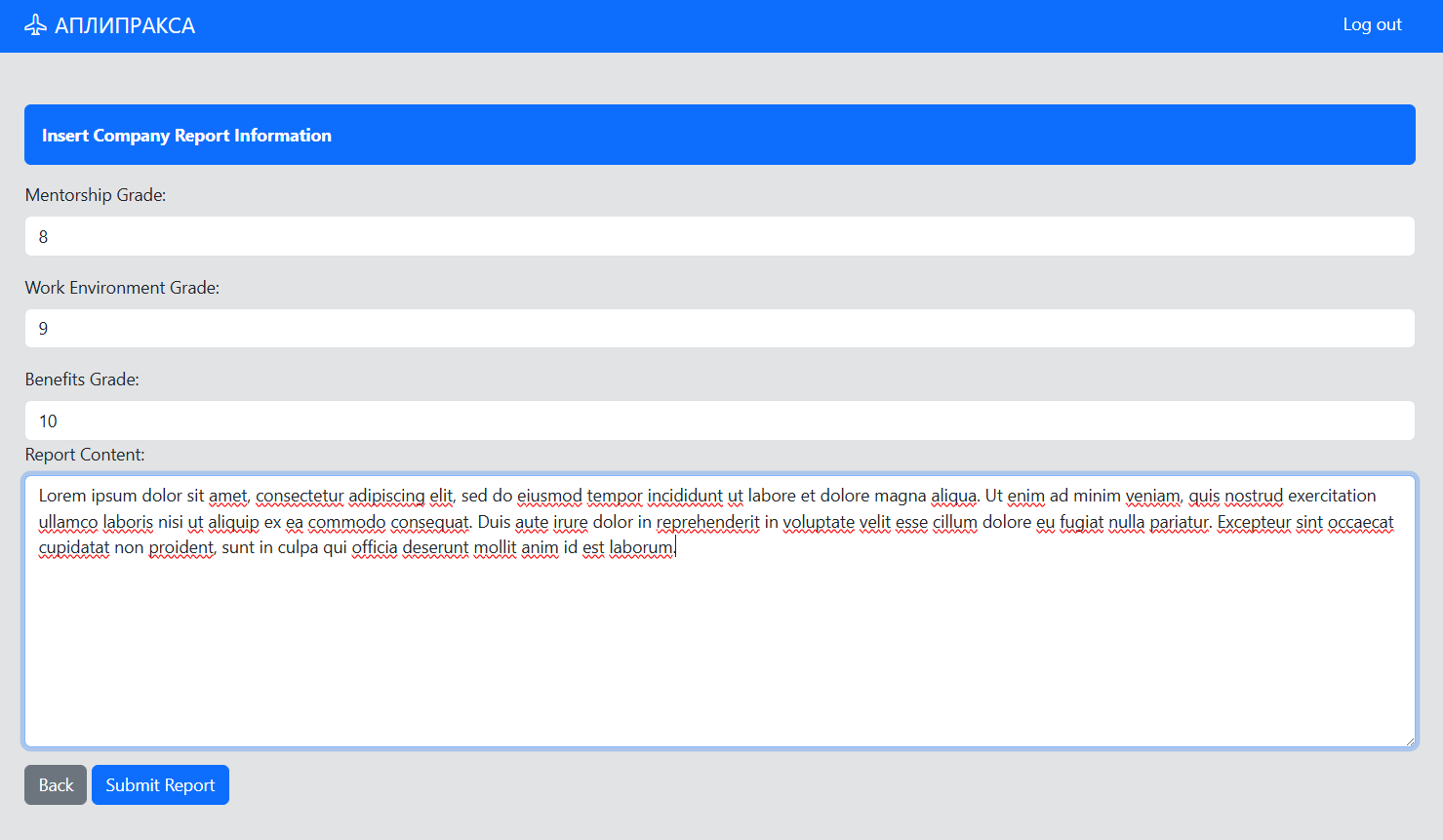
* Student Report – извештај за студентот, од компанијата
* Company Report – извештај за компанијата, од студентот



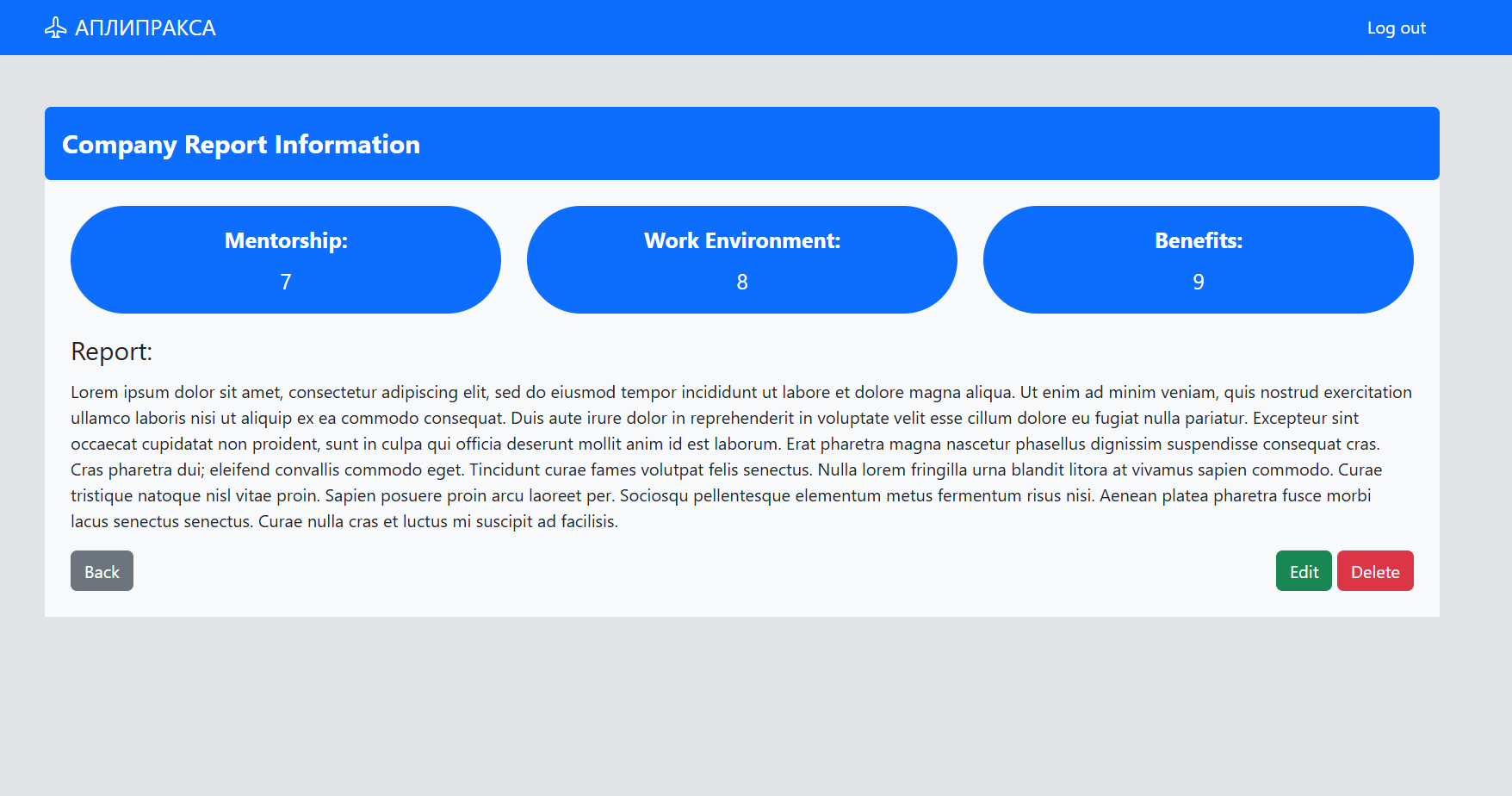
Со клик на Student Report ќе биде насочен кон друга страна каде ќе може да прочита краток извештај од страна на компанијата за неговата извршена работа. На овој извештај треба да ги има следните информации: генерална оценка, техничка оценка, комуникациска оценка, опис за студентот. Доколку сеуште не е генериран извештајот од страна на компанијата, на студентот ќе биде прикажана соодветна порака – No report has been generated yet.



Со клик на Company Report, ќе се отвори страна со симетрични функционалности, но сега во обратен правец. Доколку студентот за прв пат ја отвора оваа страна, ќе биде прикажано известување дека сеуште не е генериран извештај за компанијата. Во долниот дел на страната ќе има копче Create, кое ако се селектира, ќе се отвори форма за пополнување информации за впечатоците на студентот за компанијата. Може да се пополната следните информации: оценка за менторството, оценка за работната средина, оценка за бенефитите, опис за компанијата.



Со клик на Submit, извештајот ќе биде внесен во системот и ќе биде понатаму видлив за секоја компанија што ќе го разгледува профилот на студентот. Доколку е потребно, студентот може да ги ажурира информациите за извештајот со клик на ново-појавено копче Edit.

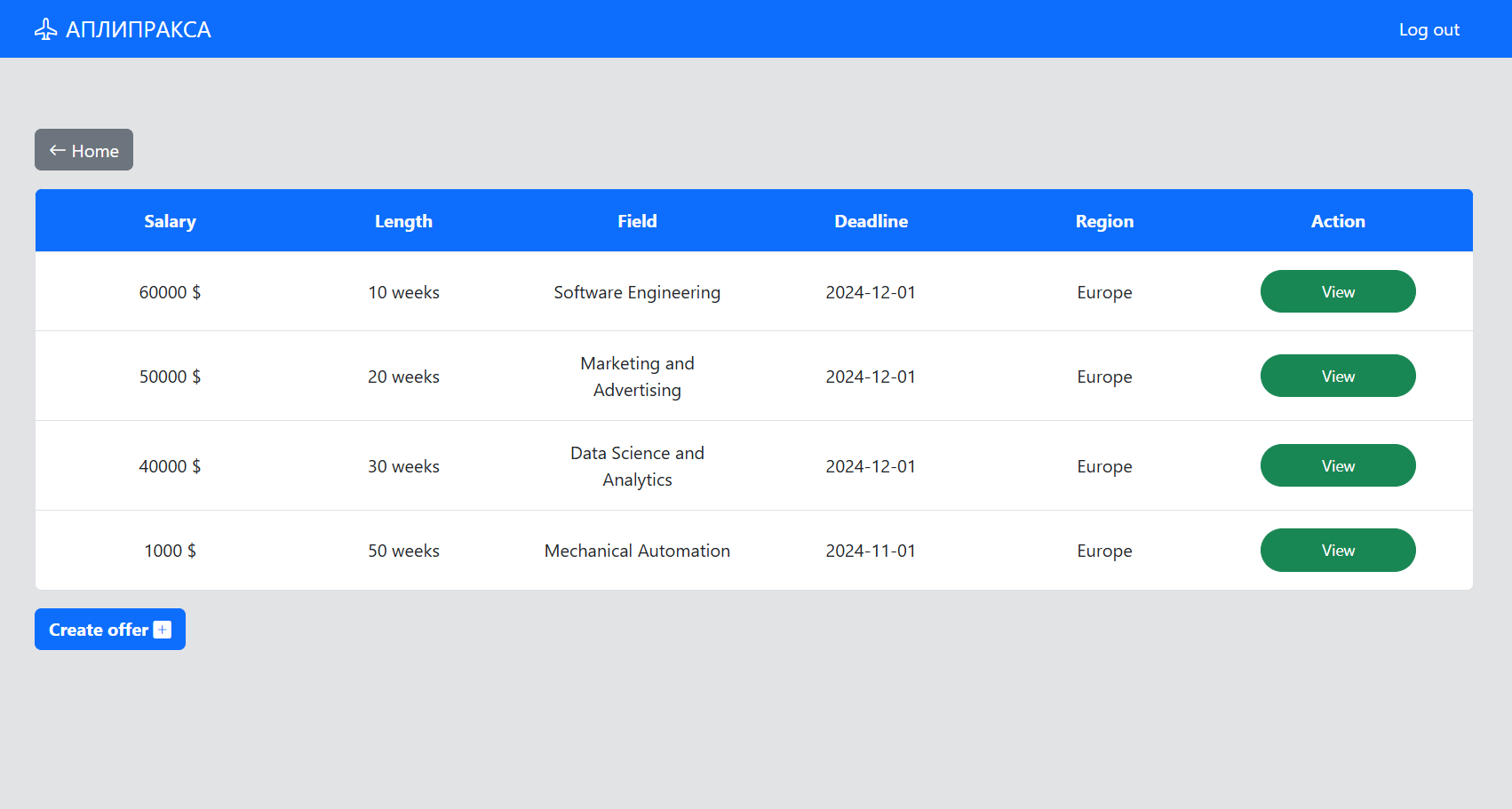


**Компании**

За корисниците компании овозможени се следните специфични услуги:

***Разгледување свои понуди***

Тргнувајќи од нивната почетна страна, компаниите можат да се пренасочат кон делот My Offers. Овде ќе биде прикажан список од сите нивни досегашно-креирани понуди за пракса, заедно со нивната кратка содржина. Секоја креирана понуда може да се отвори посебно за да се разгледа во повеќе детали.



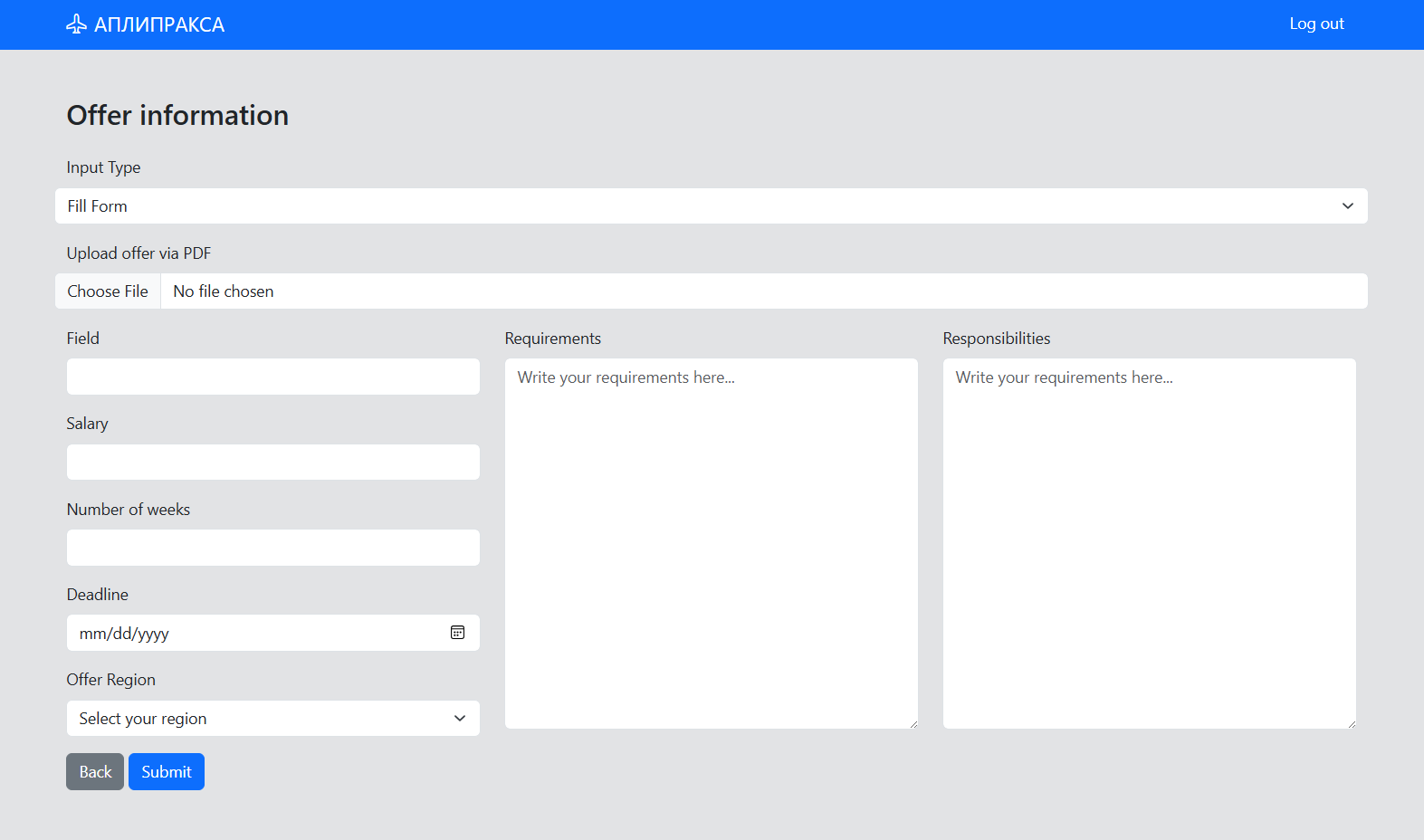
***Креирање на понуда***

Во делот за разгледување на креираните понуди, ќе има и копче Create Offe. Со клик на копчето ќе се отвори нова страна со форма за пополнување информации за новата понуда. Постојат два начини за внес на новите податоци:

* Креирање нова понуда со пополнување на формата
* Креирање нова понуда со прикачување на PDF дадотека

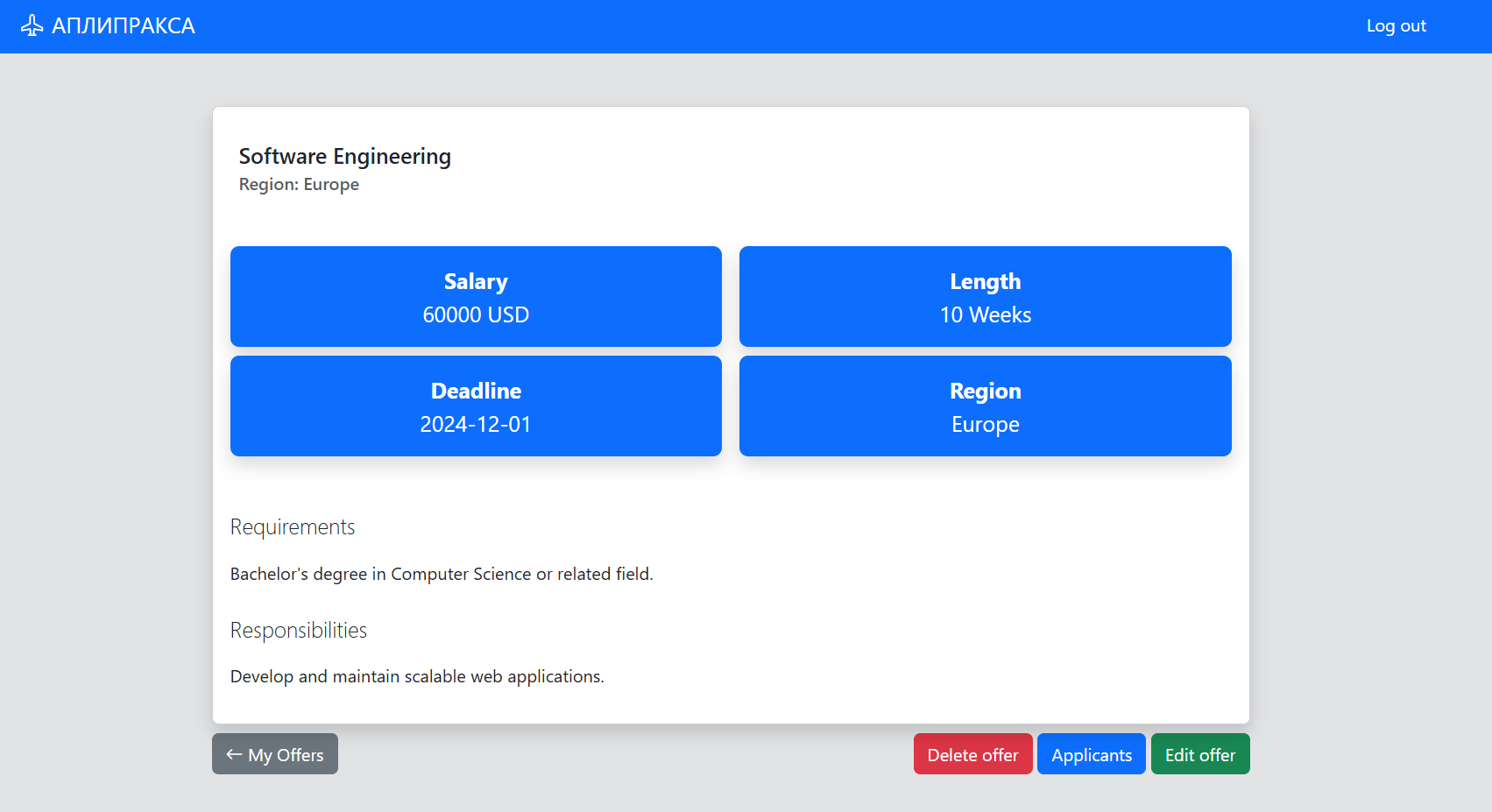
Доколку сакаме рачно да ги внесеме податоците за понудата, треба да селектираме во делот Input Type – опцијата Fill Form. Потоа во секое поле ги внесуваме потрбените информации: област, плата, број на недели, рок за аплицирање, регион, побарувања за студентот, и одговорности. На крај, мора да го кликнеме копчето Submit, по што ќе бидеме насочени кон страната со сите креирани понуди од страна на копанијата. Понудата е сега видлива за сите студенти.

Доколку сакаме да ги прикачиме информациите за понудата преку PDF дадотека, треба сега да ја селектираме опцијата Upload PDF и наместо да ја пополнуваме формата, сега можеме да кликнеме на Upload File копчето. Селектираме дадотека од нашата локална машина и ја прикачуваме. По ова, повторно ќе бидеме пренасочени кон страната со сите наши креирани понуди. За успешно креирање понуда на ваков начин, потребно е содржината на дадотеката да биде во претходно дефиниран формат – прикажан на сликата.



***Разгледување понуда посебно***

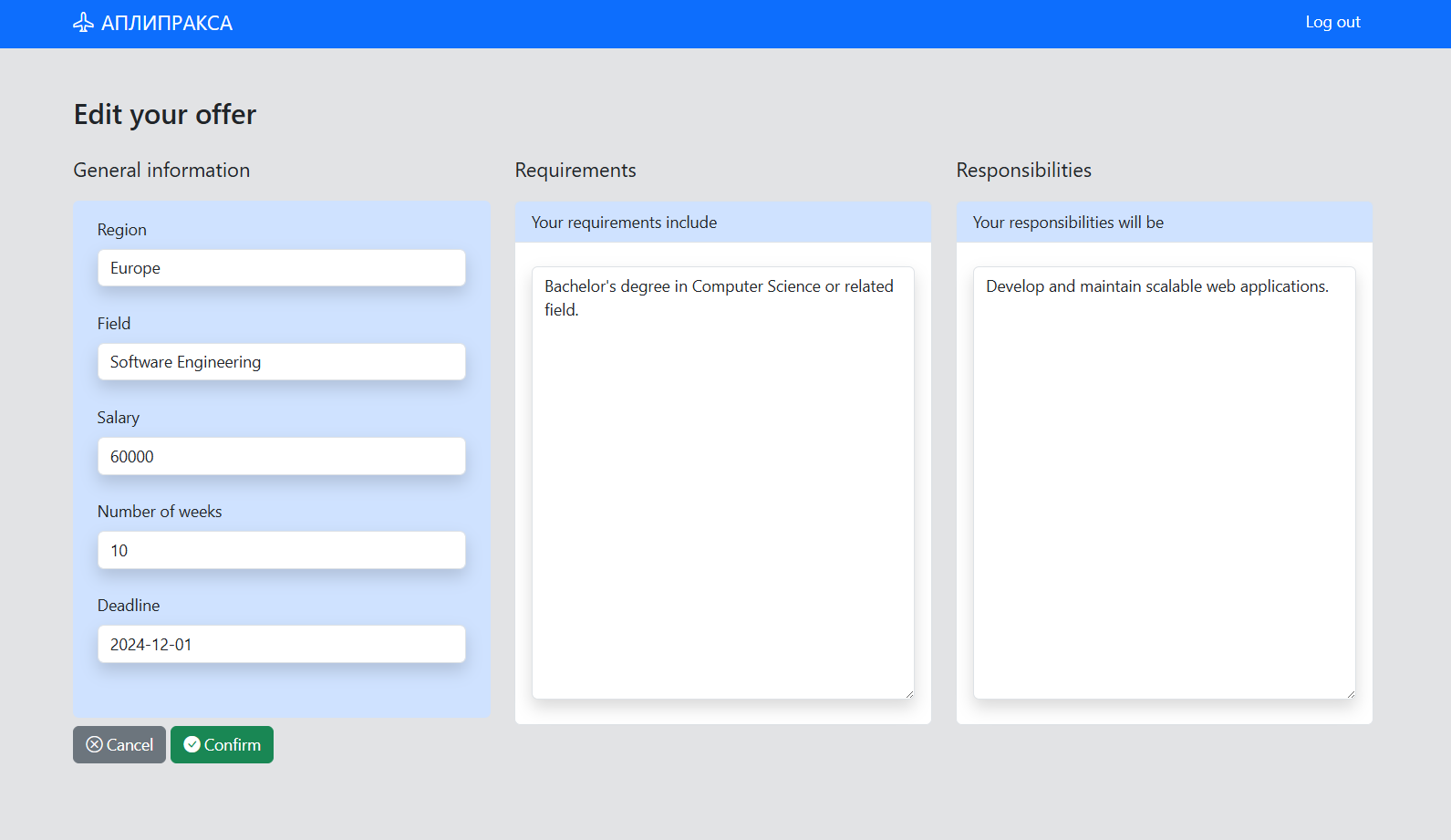
Од страната со сите наши креирани понуди можеме да ја отвориме секоја понуда посебно во повеќе детали. Ова се прави преку клик на копчето View, слично како што е за студентите. По ова, ќе се отвори страната со сите информации за понудата, вклучувајќи ги и пообемните полиња за побарувања и одговорности за понудата.



За разлика од студентите, во овој дел компаниите можат да извршуваат дополнителни акции поврзани со понудата. Конкретно, можат да ја ажурираат понудата преку клик на копчето Edit Offer или пак целосно да ја избришат понудата со клик на Delete Offer. А доколку кликната на Applicants, ќе можат да ги видат сите апликанти за понудата – за ова ќе стане збор подолу.

***Ажурирање на понуда***

Кога ќе се отвори страната за ажурирање на информациите за понудата, ќе се отвори нова страна со пополнета форма со истите информации. Корисникот може тука да промени било кое поле и откако ќе заврши со ова, може да одбере Confirm или Cancel. При Confirm, се ажурира понудата и насочени сме кон претходната страна за прикази на понудата. При Cancel, едноставно се враќаме на претходната страна.

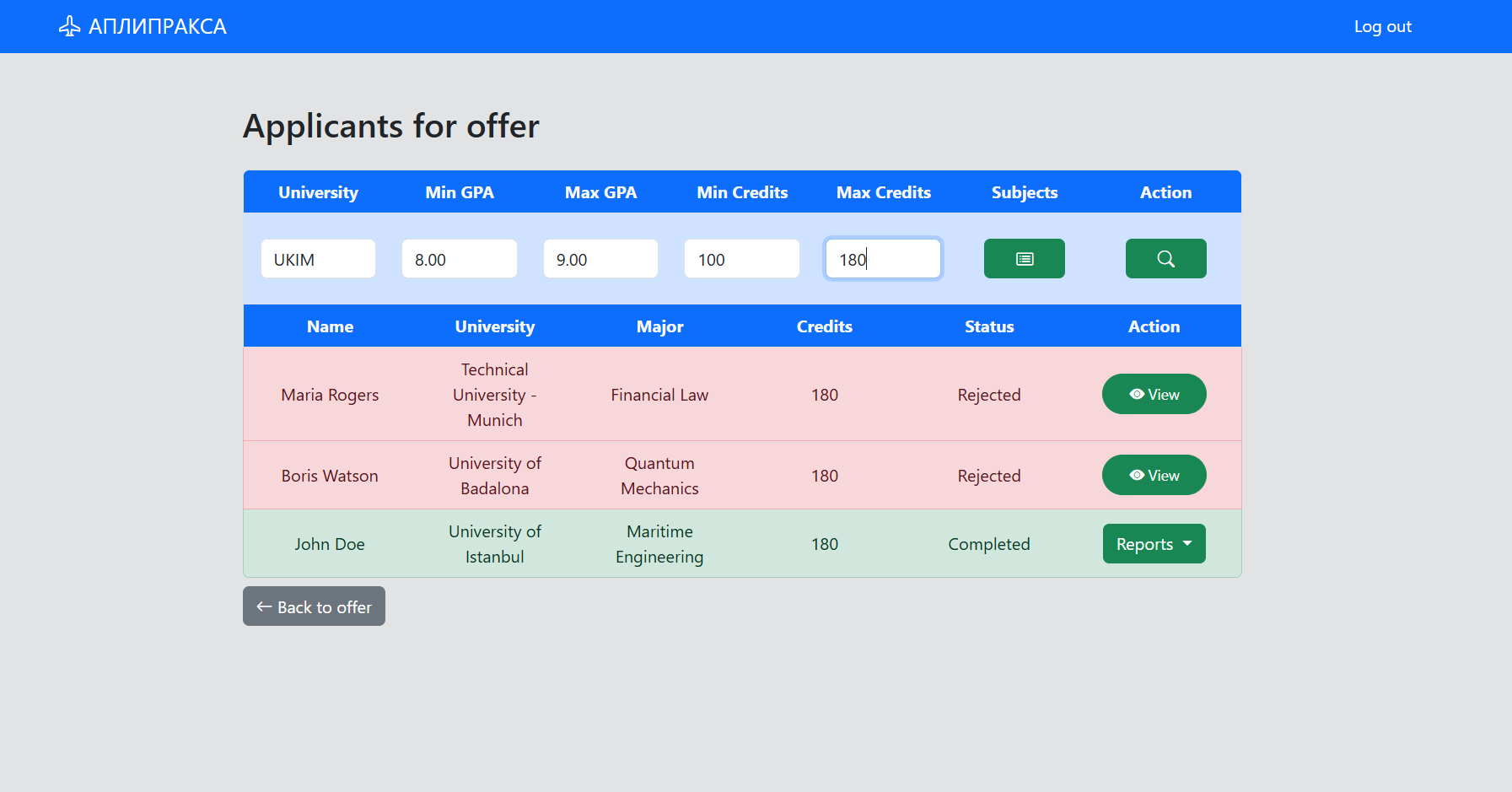


***Разгледување на апликанти за пракса***

Со пристап на Applicants делот, ќе може компанијата да види на нова страна список од сите студенти кои аплицирале за понудата, заедно со статусот на нивната апликација. Повторно, доколку апликацијата има статус Waiting ќе биде прикажана во жолта боја, доколку има статус Rejected – црвена боја, и доколку е Accepted, Ongoing, или Completed – зелена боја. За секој апликант има копче за извршување на одредена акција во зависност од статусот на апликацијата.

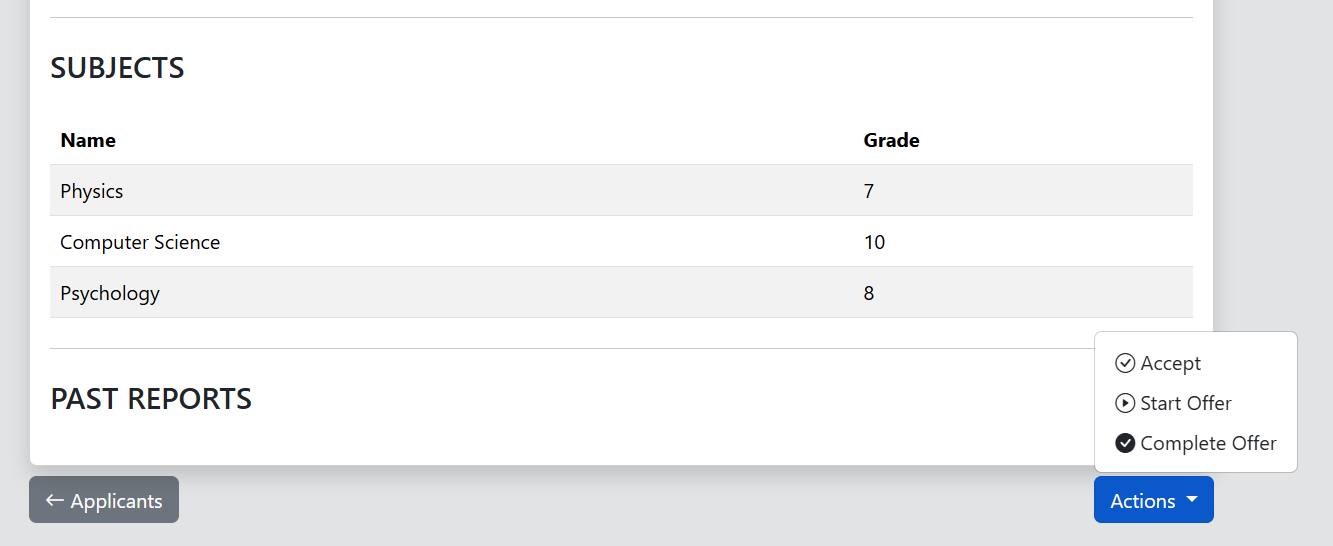
***Филтрирање на апликанти***

За да биде полесен овој процес за корисниците-компании, додадена е функционалност за филтрирање на апликантите според одредени релевантни параметри: име на универзитет, минимален и максимален просек, минимален и максимален број на освоени кредити, и положени предмети со минимална оценки. Оваа функционалност е имплементирана на сличен начин како филтрирањето на понуди кај делот за студентите. Со клик на филтер копчето без параметри, се ресетира процесот и повторно се прикажуваат сите апликанти за понудата.



***Селектирање на апликант***

Доколку компанијата сака да одбере еден студент кој ќе ја извршува понудата, тоа може да го направи така што ќе кликне на копчето View кај неговата апликација. Ќе се отвори целосниот профил на студентот. Овде ќе бидат прикажани главните кориснички информации за студентот, како и дел за неговото мотивационо писмо, неговите досегашни искуства, неговите положени предмети, и сите негови генерирани извештаи од компании каде претходно извршил пракса. Во најдолниот дел од оваа страна има копче Actions. Со клик на копчето се отвораат неколку опции: Accept, Start Offer, Complete Offer. За студент кој сеуште чека да биде прифатен, релевантна акција е само Accept.Компанијата може да ја одбере оваа акција и со ова ќе биде вратена на страната со списокот од сите апликанти за понудата, но сега статусите на апликациите ќе се променети – студентот кој бил прифатен ќе има статус Accepted, и сите останати студенти автоматски ќе добијат статус Rejected. Напомена: компанија може да прифати само студент кој има статус Waiting.

******

***Започнување на понуда***

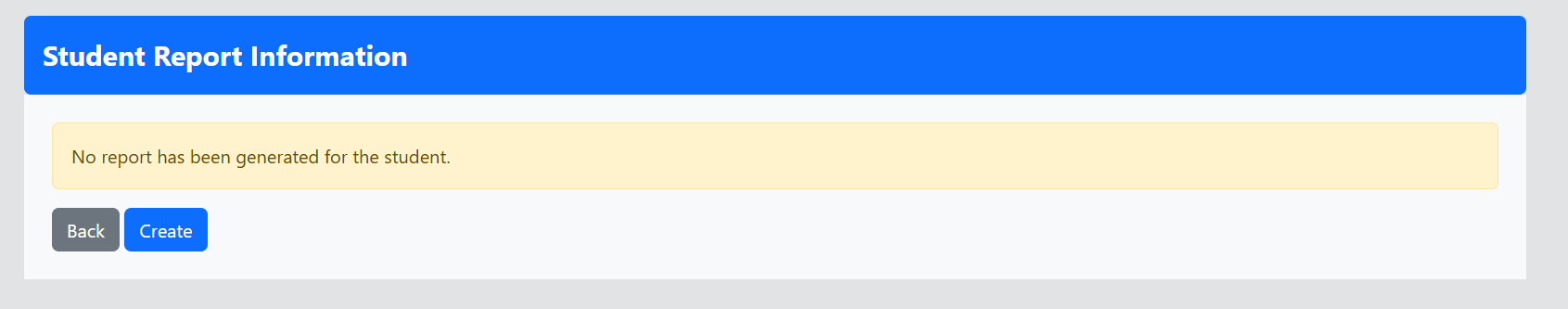
Откако одредена апликација е веќе прифатена, ќе има статус Accepted. Сега компанијата доколку ја отвори страната со списокот на сите нејзини апликанти и го отвори пак профилот на студентот, ќе може да ја селектира акцијата Start Offer. По овој чекор, ќе се отвори пак страната со сите апликанти. Сега статусот на истата апликацијата ќе биде Ongoing.

***Завршување на понуда***

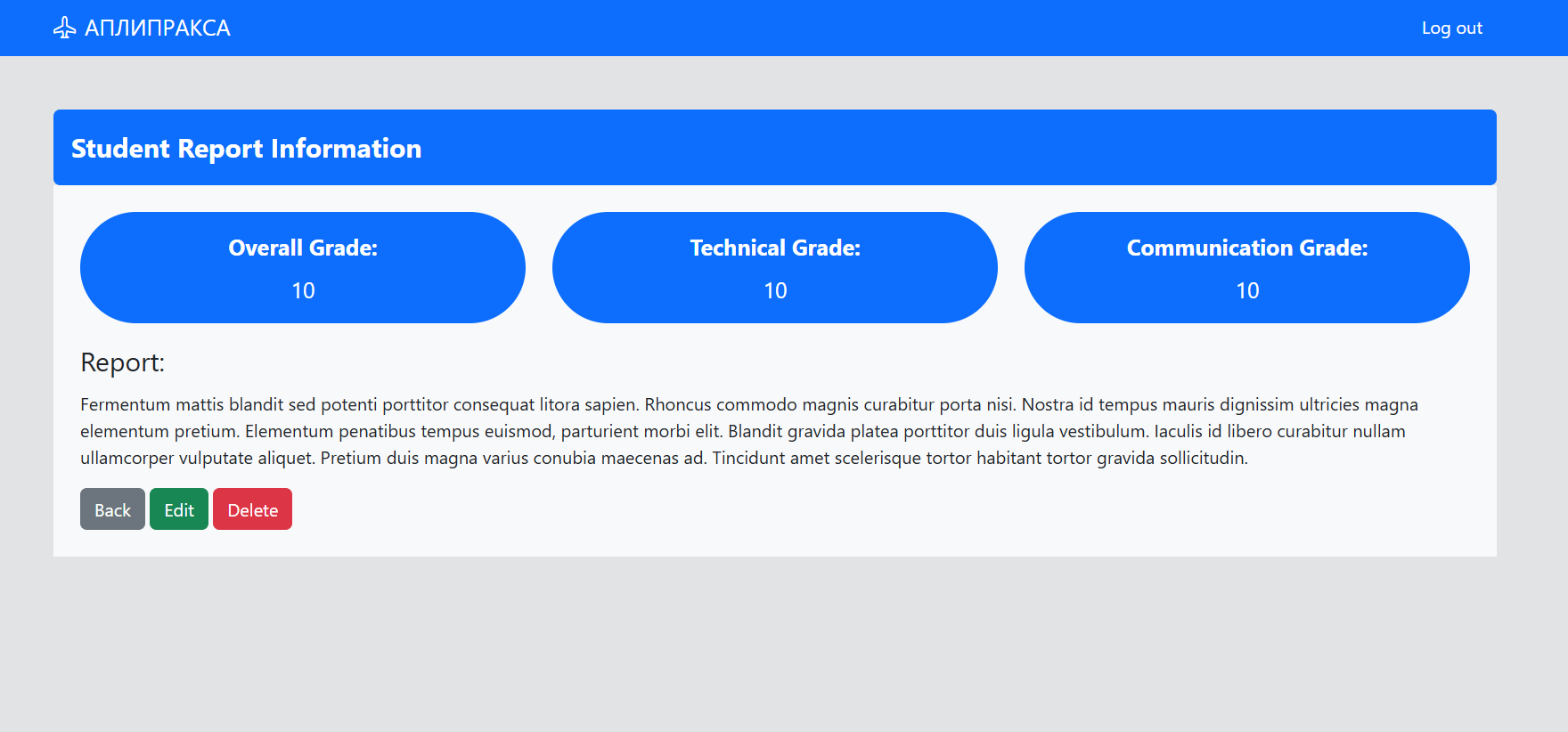
На сличен начин компанијата може да ја дефинира праксата како завршена. Ова се прави така што пак ќе го отвори профилот на студентот и ќе ја одбере опцијата Complete Offer. По ова пак ќе биде корисникот вратен на страната со сите апликации за истата понуда. Сега статусот на соодветната апликација ќе биде Completed.

***Креирање извештај за студент***

Откако ќе биде означена апликацијата како завршена, на страната со список од сите апликанти за понудата, ќе се промени копчето за извршување акција. Наместо копче View за отворање на профилот на апликантот, ќе се појави копче Reports. Со клик се отвораат две опции: Student Report и Company Report. Доколку се одбере Student Report ќе се отвори нова страна за приказ на извештај за студентот.



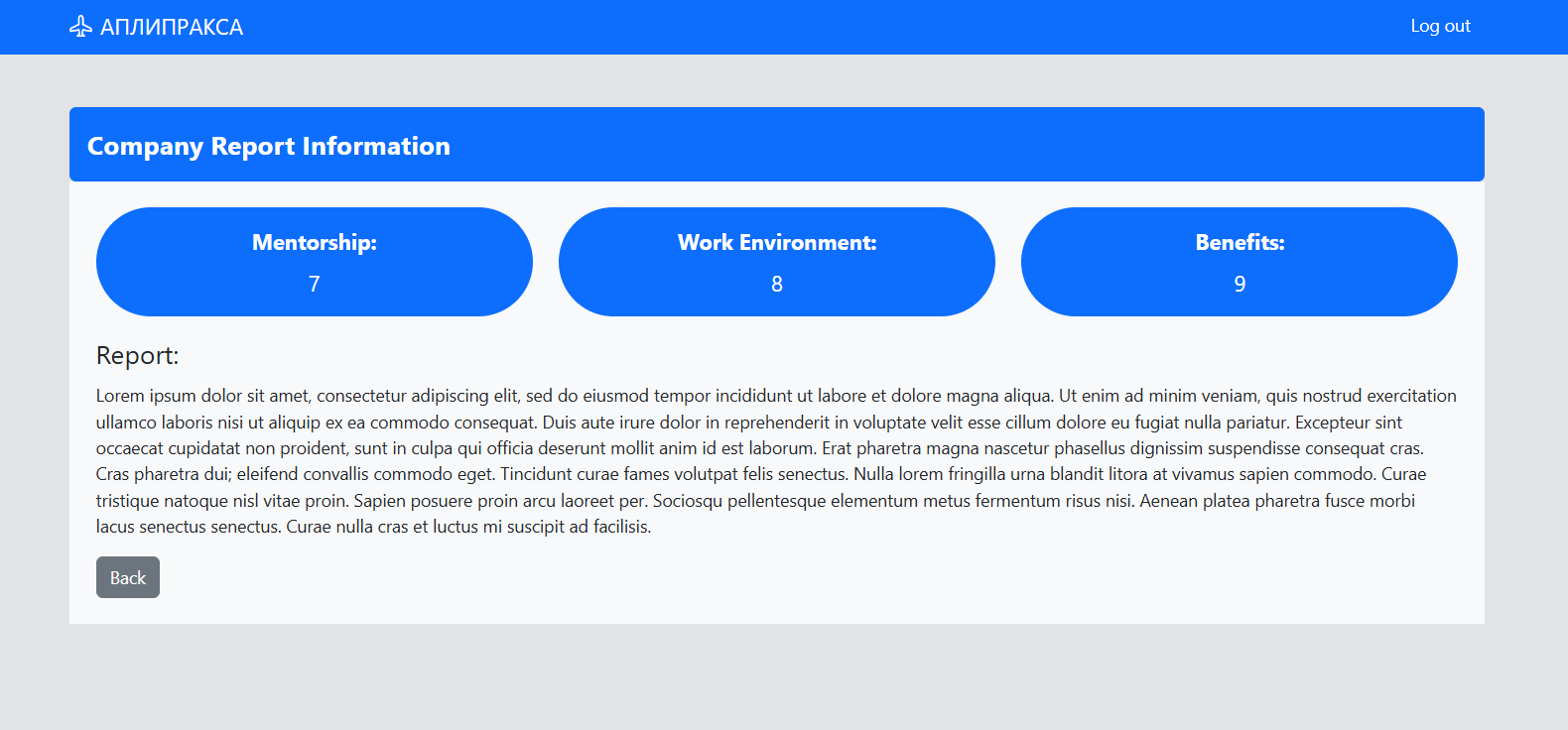
Првично нема да има никаков извештај, туку само порака до компанијата дека сеуште не е генериран извештајот. Ќе имаме копче Create. При негов клик, ќе се отвори форма за пополнување информации за извештајот за студентот: генерална оценка, техничка оценка, комуникациска оценка, и опис за студентот. При клик на Submit, извештајот се генерира и компанијата е пренасочена кон страната за приказ на извештајот.



Сега ќе се појават нови опции: за ажурирање на извештајот преку истата форма и за целосно бришење на извештајот. Компанијата од тука може и да се врати на претходната страна со списокот на апликанти.

***Разгледување извештај од студент***

Доколку копанијата кликне на Company Report, ќе се отвори нова страна за приказ на извештај за компанијата кој треба да е генериран од студентот кој ја извршил истата пракса. Доколку сеуште нема студентот креирано извештај, ќе биде прикажана соодветна порака – No report has been generated yet. Секако, бидејќи студентот е одговорен за овој извештај, компанијата во овој чекор нема да има опции за креирање, ажурирање и бришење на извештаи. Доколку извештајот е креиран од студентот, ќе бидат прикажани следните информации: оценка за менторство, оценска за работна средина, оценка за бенефити, опис за компанијата.



Извештаите се со цел понатамошни студенти и компании да имаат поголем увид во профилот и резимето на нивните потенцијални практиканти и работодавци.

**Регионалност**

Во рамки на сите претходни функционалности, вградена е и логика за користење на регионалноста на студентите и праксите. Студентот при разгледување на понудите за пракса ќе може само да гледа понуди кои се во неговиот регион или пак тие што се означени како глобални понуди. За останатите понуди ќе му биде оневозможен пристапот. Во рамки на системот се вградени следните региони: Европа, Азија, Америка, и глобален регион. Но извршување на промени – додавање и менување на кои региони се во системот – е релативно лесна операција која се состои од промена на одредена конфигурација на ситемот. Но ова не е функционалност овозможена за корисниците.

**Меко бришење**

За да се избегне проблемот на трајно губење на податоци кои можат да бидат клучни во развојот на апликацијата имплементирано е во некои делови “меко бришење” на податоците. Конкретно, доколку компанија сака да избрише одредена понуда за прака, таа може да го направи тоа, и повеќе нема да се прикажува таа понуда никаде во системот. Но во позадина, таа понуда само ќе биде тргната од досегашниот корисник – сеуште ќе се чува во базата на податоци.

**ДЕЛ 2: ТЕХНИЧКА ПЕРСПЕКТИВА**

**Кои технологии се искористени?**

***PostgreSQL***

PostgreSQL е моќен, open-source, realational database management system кој е познат за неговата доверливост, комплетност на функционалности, и совпаѓање со стандардите за складирање и работење со податоци. Поддржува и понапредни функционалности како што се JSON податочни типови, комплексни SQL прашалници, индексирање, и можност за проширување на функционалностите преку кориснички дефинирани функции. Во контекст на веб апликација, PostgreSQL се користи за складирање и менаџирање со структурирани податоци како што се кориснички профили, и за безбедно извршување на трансакции. Неговиот капацитет за работење со голем обем на податоци на ефикасен начин, го прави одлична технологија за апликации кои имаат потреба од скалабилност и робустно работење.

Во нашата апликација, сите податоци се складираат во PostgreSQL база. За комуницирање со базата и извршување на сите SQL прашалници искористен е python драјверот – psycopg. Дополнително користиме две бази на податоци кои се идентични по нивната струкутура – еден за вистинско работење со апликацијата, а друг за тестирање.

***Python***

Python е генерален програмски јазик на високо ниво кој е познат се неговата читливост и богат екосистем на библиотеки и framework-ови. Се користи во голема мера за равој на веб апликации, машинско учење, автоматизација и скриптирачки процеси. Во контекстот на веб апликации, Python служи како програмски јазик за развој на beckend логиката и им овозможива на софтверските инженери да имплементираат апликациска бизник логика, да извршуваат комуникација со базата на податоци и менаџирање на останатите операции на страна на серверот. Неговата најголема предност е што може да се искористи за релативно брз развој на функционалности и промена на бизнис логика доколку е потребно.

Нашата апликација го користи Python во голема мера, и тоа за креирање на сите рути, контролери, конекции со базата, безбедностни функционалности и голем број на други помошни операции. Исто така го искористивме Python за креирање и извршување автоматизирани тестови за нашата апликација – за ова ќе наведеме повеќе детали подолу.

***FastAPI***

FastAPI е модерен, Python-базиран веб framework дизајниран за градење на API-ња на брз и ефикасен начин. Поддржува асинхроно програмирање, автоматско валидирање на влезните и излезните податоци користејќи ги type-hint-овите на Python. Нуди и автоматски генерирани интерактивнa API документација користејќи ja OpenAPI спецификацијата. Во контекстот на веб апликации, FastAPI се користи за справување со HTTP повици, имплементирање на бизнис логика и да изложува точки на пристап преку кои може да се комуницира со базата на податоци и останати сервиси. Неговата брзина и леснотија при користење го прави одличен избор за градење микросервисна архитектура и RESTful API-ња.

Во нашата апликација, FastAPI е искористен за креирање на сите рути и податочни шеми за сите наши влезни и излезни податоци, како и за креирање на локален сервер кој може да го пристапуваме со нашите HTTP повици.

***Jinja Templates***

Jinja е алатка за генерирање на темплејти за Python. Овозможува креирање на динамични HTML страни така што преку вградување на логички експресии слични на Python во самата HTML страна. Може да се искористи за генерирање if-else логика, for циклуси, наследување од други генерални темплјети, реискористување на компоненти преку функции и слични функционалности. Во контекстот на веб апликации, Jinja се користи за генерирање на сите погледи и приказот на корисничките податоци. На пример, FastAPI апликација може да рендерира Jinja темплејт за приказ на кориснички профили, извештаи и сл.

Во нашата апликација Jinja е клучен во приказот на сите наши информации, особено кога се во прашање динамичен број на податоци, како што се нашите понуди за пракса, апликациите, и корисничките извештаи. Во одредени делови се користи и за реискористување на одредени компоненти како што се разните копчиња што ја поврзуваат целата наша апликација. Го искористивме најмногу за логиката за апликациите, приказот на различни бои, и за приказот / сокривањето на одредени функционалности во зависност од типот на корисникот.

***JavaScript***

JavaScript е интерпретиран програмски јазик на високо ниво, кои се користи воглавно за развој на клиентска логика за веб апликации, и за интерактивни и динамични веб страни. Во веб апликации, се справува со кориснички интерфејси, ажурирање на содржината на страната без да се изврши refresh на страната, и нуди беспрекорно интегрирање со API-ња. На пример, FastAPI може да опслужува backend податоци во JSON формат, и JavaScript може да ги конзумира тие податоци за да ги ажурира корисничките интерфејси во моментално (real time) време.

Во нашата апликација JavaScript користиме при секоја транзиција од една во друга страна, односно за извршување на секој HTTP повик до FastAPI рутите. Го користиме JS и за зачувување на клучни информации кои се потребни на корисниците за пристапување на целата апликација – конкретно преку JS ги зачувуваме токените кои го автентицираат и авторизираат корисникот при секоја акција. Се користи и за зачувување на податоци за моменталната корисничка сесија – offer id, student id, company id и слични податоци.

***Bootstrap & HTML***

Bootstrap е популарна front-end технологија за градење респонсивни веб страни. Нуди предефинирани CSS класи, Javascript компоненти, систем на редици и колони, и слични елементи што овозможува на софтверските инженери брз развој на визуелно привлечни кориснички интерфејси. Заедно со HTML, што е стандардниот markup јазик за структурирање на веб содржини, овозможуваат на девелопери да креираат чисти, професионални и приспособливи кориснички шаблони. Веб апликациите често го користат Bootstrap за додавање на визуелни ефекти на форми, копчиња, навигациски секции. Сето ова се прави во рамките на HTML содржината на веб страниците.

Во нашата апликација Bootstrap е искористен во сите темплејти, за креирање на стилот на нашите форми, списоци, апликации, и извештаи. Само во одредени делови искористен е чист CSS за креирање на специфичен дизајн на одредени компоненти, и тоа само кога не постоела соодветна Boostrap класа.

***Pytest & Bash***

За креирање на авотматизирани тестови – за проверка на функционалностите на апликацијата, ги искористивме технологиите: Pytest и Bash.

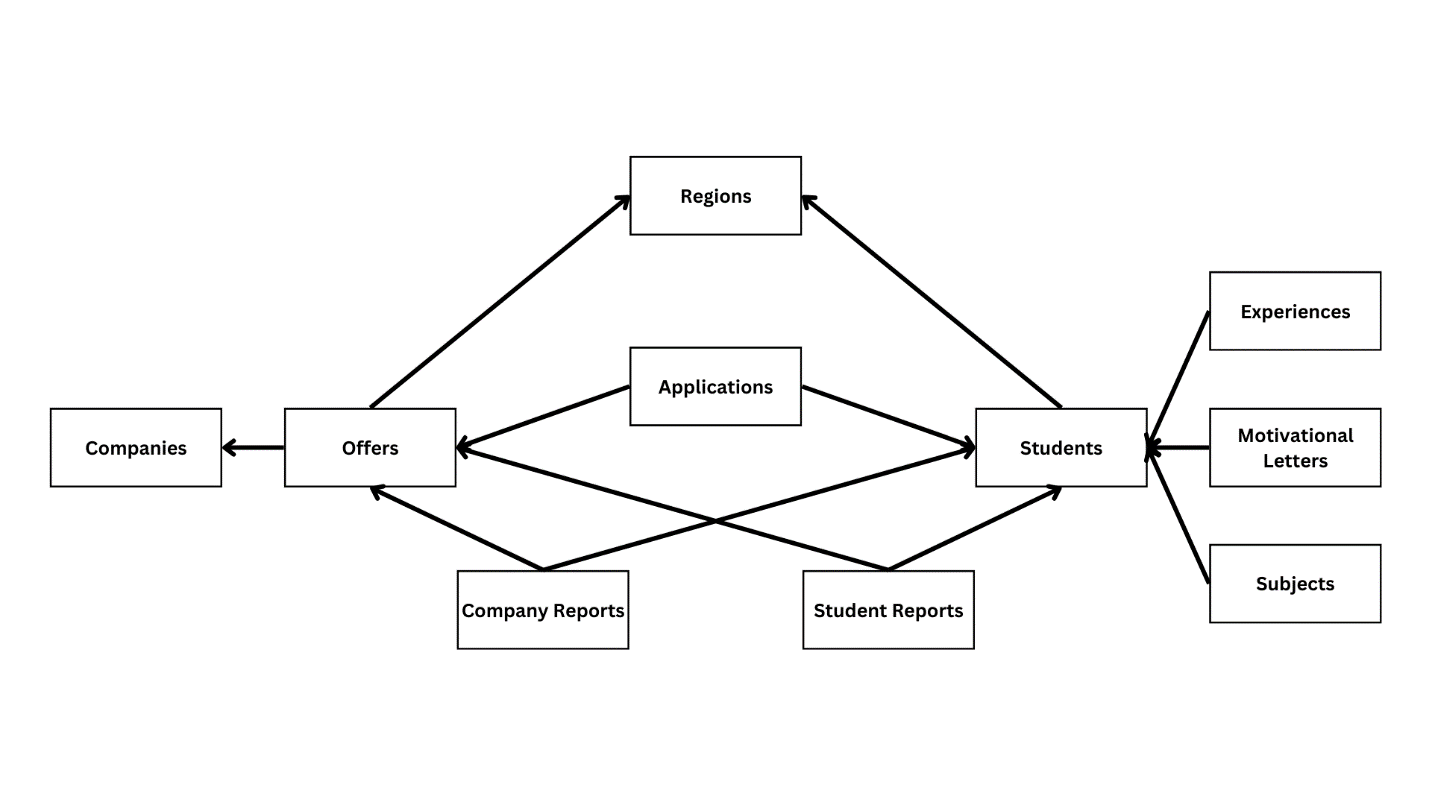
Pytest е Python библиотека за тестирање што го поедноставува процесот на пишување, извршување, и организирање на unit тестови, интеграциски тестови, и end-to-end тестови. Во контекст на веб апликација, Pytest може да се користи за тестирање на: точките на пристап на апликацијата, прашалниците до базата на податоци, и апликациската логика, за да потврди дека софтверот работи како што би очекувале. Содржи fixtures и plugin-и кои го прават доста флексибилен и соодветен за целосно автоматизирање на тестови.

Bash е скриптирачки јазик кој се користи за автоматизација на задачи во развојната околина. Во веб апликации, Bash скрипти често се користи за сетапирање на сервери, распоредување на апликации, извршување на тестови, и сетапирање на конгурациски информации за конектирање со базата на податоци.

Во нашата апликација, имаме креирано повеќе end-to-end тестови за проверка на работењето на нашите endpoints и дали правилно се креираат, ажурираат, читаат и бришат нашите ентитети. Конкретно имаме тестови за функционалностите за студентите, компаниите, понудите за пракса, апликациите, и работните искуства на студентите. Но за автоматизирано извршување на тестовите и за креирање на соодветна конекција со тестирачка база на податоци, користиме Bash скрипта.

**Каква е структурата на базата на податоци?**

Нашата база на податоци ја има следната структура:



Ги користиме следните табели за складирање на податоците:

**Региони:** Конфигурациски податоци за регионите во системот. Можат да се прошират, но генерално би требало да се користат како статички податоци. Моменталната верзија на апликацијата содиржи 4 региони: глобален регион, Европа, Азија, и Америка.

**Студенти:** Сите главни кориснички информации за студентите – и за најавување на системот, и за приказ на целосниот профил на студентот. Ова е една од клучните табели во нашиот ситем. Може да се види дека има најголем број на релации со други табели.

**Искуства:** Информации за поранешните искуства на студентите.

**Предмети:** Информации за досегашните положени предмети на студентите заедно со нивните оценки.

**Мотивационо писмо:** Информаци за мотивационите писма на студентите. Ги има следните секции: About Me, Skills, Looking For

**Компании:** Информации за корисничкиот профил на компаниите. Се користат за најавување на системот и за приказ на целосниот профил на компаниите.

**Понуди:** информации за понудите кои се креирани од страна на компаниите.

**Апликации:** Информации за сите апликации од студентите за различните понуди.

**Извештаи за студенти:** Информации за извештаите креирани од компаниите за студентите кои извршиле пракса кај нив.

**Извештаи за компании:** Информации за извештаите креирани од студентите за компаниите кај кои извршиле пракса.

Ќе објасниме повеќе околу релациите што постојат во базата:

**Студент – Регион (М-1):** Секој студент мора да има дефинирано од кој регион е. Ова се дефинира при регистрација на студентот.

**Понуда – Регион (М-1):** На ист начин, мора секоја понуда да има дефиниран регион. Ова се прави при креирање на понудата и може да се промени понатаму.

**Искуства – Студент (М-1):** Секој студент може да има различни искуства поврзани со неговиот профил.

**Предмети – Студент (М-1):** Секој студент може да има различен број на предмети кои ги положил.

**Мотивационо Писмо – Студент (опцинална 1-1):** Секој студент може да има мотивационо писмо. Но ова не е задолжително за студентите.

**Понуди – Компанија (М-1):** Секоја компанија може да креира произволен број на понуди.

**Понуда – Апликација – Студент (М-N):** Секој студент може да аплицира на произволен број на понуди и секоја понуда може да има произволен број на апликанти.

**Понуда – Извештај за Студент – Студент (1-1 и М-1):** За секоја извршена пракса може да имаме извештај за студентот. За секој студент може да има повеќе извештаи. Но за една понуда може да има само еден студентски извештај (само еден студент може да ја заврши понудата).

**Понуда – Извештај за Компанија – Студент: (1-1 и М-1):** За секоја извршена пракса може да имаме извештај за компанијата. Еден студент може да е автор на повеќе извештаи. Но извештајот може да се однесува на само една понуда (само студентот кој ја завршил праксата може да остави извештај).

**Каква е структурата на кодот?**

Кодот на апликацијата е структуриран на следниот начин:

* Source: Ја содржи back-end бизнис логиката во Python и FastAPI.
* SQL: Ги содржи сите SQL изрази за креирање на табелите, домените, функциите на ниво на базата.
* Static: Ги содржи сите статички фајлови потребни за front-end логиката т.е. сликите и JavaScript кодот
* Templates: Ги содржи сите Jinja темплејти кои се користат за серверско рендерирање на страните.
* Tests: Ги содржи сите Pytest тестови
* Помошни дадотеки: Поставени се во главниот фолдер на проектот. Служат за помош при процесот на развој и тестирање на апликацијата.

***Source***

Понатаму source кодот е организиран во следните дадотеки:

* Main: Содржи логика за започнување и завршување на апликацијата, вчитување на сите конфигурациски информации, поставување конекција со базата на податоци, и вчитување на фолдерот за фотографии. Ова е првата скрипта која се повикува при започнување на апликацијата.
* Database: Содржи логика за вчитување на променливите за конекција со базата на податоци, креирање на конкциски string, како и креирање на конекциски базен (connection pool) кој се повикува во останатите делови од системот.
* Schemas: Ги содржи сите влезни и излезни податочни шеми кои понатаму се користат од FastAPI за поставување на правилата за какви HTTP повици и одговори се дозволени во системот. Има по една шема за секој ентитет (табела) во системот и скоро по една за секоја од CRUD операциите. За одредени услуги има и шеми кои комбинираат полиња од различни ентитети, на пример за целосниот профил на студентот.
* Routes: Ги содржи дефинициите на сите влезни точки (рути) на апликацијата. Ги користи претходните шеми за дефинирање на влезните и излезните податочни структури. На пример, за креирање на на нов студент ја користи StudentCreate шемата.
* Controllers: Го содржат јадрото на апликациската логика. Повикани се од нашите рути. Ги извршуваат сите проверки за валидностна на барањата на корисниците, и ги извршуваат соодветните промени и прашалници во базата на податоци.
* Queries: Ги содржат сите SQL изрази за читање, креирање, промена, и бришење на податоци. Во некои делови постои логика за динамично генерирање на SQL изрази во зависност од потребната – на пример, за филтрирање на апликантите, се променуваат филтер-параметрите, зависно од барањето на корисникот. Се користат воглавно од страна на претходните контролери.
* Enums: Содржи одредени категоризации на поврзани со апликациската логика. Конкретно содржи категоризации за: статус на апликациите, типот на корисник, регионот, како и конфигурациските податоци (од .env дадотеката). Се користат насекаде низ апликацијата.
* Notifications: Ја содржи целата апликациска логика за испраќање на нотификации до корисниците по е-пошта. Се повикува од страна на рутите.
* Profile Pictures: Ја содржи целата апликациска логика за справување со корисничките слики, односно зачувувањето на сликата во соодветен фолдер, ажурирање на патека до сликата во базата, промена на сликата, промена на патеката до сликата, бришење на претходната слика, како целосно бришење на слиаката заедно со патеката до неа. Ги повикува SQL изразите од queries-делот како и шемите за проверка на типон на корисникот.
* Files: Ја содржи целата логика за извлекување на податоци за пракса при креирање на понуда преку прикачување на PDF дадотека. Повиквана е оваа скрипта во контролерите.
* Security: Ја содржи целата безбедностна логика. Содржи функционалности за креирање на JWT токени, декодирање на токени, одредување на корисникот-сопственик на токенот, автентикација на корисник, авторизација, читање на корисник врз основа на е-пошта. Ги повикува шемите и помошните функции.
* Utils: Содржи мал број на помошни поопшти функции – за екстрахирање инфомации за положените предмети од текстуално поле и за одредување на тип на корисник – студент или компанија.

Дел 3 - Понатамошен развој

...

Што може да се подобри?

...

Што може да се додаде?

...

Каков бизнис модел може да се имплементира?

...

Референци

* Python docs
* FastAPI docs
* Jinja Template docs
* PostgreSQL docs
* Bootstrap docs
* Pytest docs
* Effective Python
* The Pragmatic Programmer
* Python Tricks
* Mozilla Developer Docs
* Real Python blog