

# 

Проект по предметот „Имплементација на системи со отворен и слободен код “

RoomRevamp

Изработил:

Виктор Тасевски

191149

Професори:

д-р Јоксимоски Бобан

д-р Чорбев Иван

Содржина

[**Вовед 4**](#_inwh82hrd5eo)

[**Спецификација на софтверски барања 5**](#_mvqqhxh3hvig)

[1. Управување со корисници 5](#_fu6dcqqzf7qn)

[1.1. Рољи на корисници 5](#_z4rr96q9142m)

[1.2. Регистрација на корисници 5](#_9mts72co3rbw)

[1.3. Автентикација на корисници 5](#_ev32xn83mapo)

[1.4. Заборавена лозинка 6](#_sogrjlgw49bv)

[2. Управување со проекти 6](#_uw8f16dgxmw4)

[2.1. Создавање и ажурирање на проекти 6](#_man9h9pa2v4u)

[2.2. Преглед на проекти 6](#_2h2sixwxfd6g)

[3. 3D модели 6](#_8tw1nyyvgtey)

[3.1. Поставување на 3D модели 6](#_bheqm8lmvbbq)

[3.2. Ажурирање и бришење на 3D модели 6](#_f1b1g4gl8y3)

[3.3. Преглед на 3D модели 6](#_cr157auz7lp3)

[4. Сигурност и автентикација 7](#_u2188ianucdc)

[**Технологии и алатки 7**](#_ennzqyej0cao)

[Фронтенд технологи 7](#_4u4dhs5vx9p)

[Бекенд технологии 7](#_upiepbnvp1yr)

[Меил сервис 8](#_sbjm5wuhs56m)

[База на податоци 8](#_jvr3hybn950d)

[Контрола на верзии и хостинг 8](#_ftzu4gfhkclx)

[Алатки за развој 8](#_am1aidenfkva)

[Континуирана интеграција и испорака (CI/CD) цевковод 8](#_v3urxs1mkaes)

[Фази на CI/CD цевководот: 9](#_yoylx6liadt8)

[**Функционалности 10**](#_4gs27sldjp88)

[Регистрација на корисник 11](#_nu8xxclli540)

[Најава на корисник 13](#_f9kbfynxb9rt)

[Заборавена лозинка 15](#_yetvme3ygxv6)

[Создавање на нов проект 18](#_a0ea7uie09yi)

[Преглед на проекти 19](#_oa2dcosoi49g)

[Ажурирање и бришење на проект 19](#_5lp77k1buvsj)

[Уредување на ентериерот 20](#_uzslabl9irf2)

[Додавање на 3D модел 23](#_lf2x6qjs1ihw)

[Преглед на 3D модели 24](#_4qz7ft4jsj20)

[Ажурирање и бришење на 3D модел 25](#_8off5wgotgbq)

[Промена на лозинка 26](#_peidotck5fsn)

[Одјававување 27](#_3367tzah9dg4)

[**Стартување на апликацијата 28**](#_23i3f23sehne)

[Начин 1 28](#_sgjs3jr5i787)

[Стартување на бекенд 28](#_sjuwfaeueb0c)

[Стартување на фронтенд 28](#_fxdse9hzon33)

[Начин 2 29](#_u8u9h8cryp0v)

[Стартување на бекенд, фронтенд и база 29](#_byvmc2cscu96)

[Начин 3 29](#_jiavpr6i3jm9)

[Стартување на база 29](#_1vheze43u6xu)

[Стартување на бекендот 29](#_inn7rgq5901b)

[Стартување на фронтенд 30](#_ik4rg0zgmnx5)

[Влез во нашата апликација 30](#_6tczvrv5vlu)

# 

# **Вовед**

Во ера дефинирана со технолошки напредок и растечки акцент на персонализирани животни простори, ентериерниот дизајн се развил во фузија на уметност, наука и технологија. Сливот на овие елементи ја создаде новата парадигма во ентериерниот дизајн, која се обидува да го револуционира начинот на кој луѓето ги перцепираат и интерактираат со нивните околини. Во одговор на оваа развивачка сцена, јас се впуштив во патување за создавање на најновата веб апликација која го олесну ентериерниот дизајн на интуитивен и имерзивен начин.

Mojata веб апликација, изградена на стабилна основа на Laravel, React.js, Three.js и PostgreSQL, претставува значаен напредок во технологијата на ентериерниот дизајн. Таа ја искористува моќта на модерните алатки за веб развој и универзалноста на релациониte бази на податоци за да обезбеди платформа за дизајнери на ентериер, сопственици на домови и ентузијасти. Овој извештај за проектот служи како целосна документација за нашата работа, која детално ги опишува целите, методологијата и карактеристиките на нашата веб апликација.

Цели:

Примарната цел на овој проект е да се развие веб апликација која ќе го оптимизира ентериерниот дизајн процес, го прави подостапен, интерактивен и ефикасен за корисниците. Конкретно, нашите цели вклучуваат:

* Подобрување на корисничкото искуство: Целта е да се создаде ангажирачки и лесен за користење интерфејс кој овозможува на корисниците да визуализираат и експериментираат со ентериерните дизајн концепти без напор.
* Ефикасно управување со податоци: Имплементација на стабилен систем за бази на податоци (PostgreSQL) за складирање и управување со податоци поврзани со дизајн, обезбедувајќи интегритет и надежност на податоците.
* Имерзивна 3D визуализација: Искористување на Three.js, моќна 3D графичка библиотека, за обезбедување на корисниците со динамична и имерзивна 3D визуализација на нивните ентериерни дизајн концепти.
* Скалабилност и одржување: Дизајнирање на архитектурата на апликацијата за скалабилност и можност за идни подобрувања на ум.
* Безбедност и приватност: Поставување на безбедност и приватност на податоците на корисниците како прв приоритет за изградување на доверба меѓу корисниците.

Во наредните делови од овој проектен извештај, ќе се навлеземе подетално во методологијата која беше користена во процесот на развој и ќе ги истражиме клучните карактеристики на нашата веб апликација.

Оваа веб-апликација претставува значајна пресвртница во пресекот на дизајнот на ентериер и технологијата и се надеваме дека ќе ги поттикне корисниците да ја ослободат својата креативност, да ги трансформираат нивните животни простори и да ја подигнат уметноста на дизајнот на ентериер на нови височини.

# **Спецификација на софтверски барања**

### **Управување со корисници**

#### **Рољи на корисници**

* RoomRevamp подржува две рољи на корисници: Администратор И Обичен Корисник.
* Администраторите имаат дополнителна привилегија, како можноста за поставување на 3D модели кои ќе се споделат со сите корисници.

#### **Регистрација на корисници**

* Корисниците може да се регистрираат со своите е-маил адреси.
* По успешно регистрирање се испраќа линк на е-маил адресата на корисникот за верификација на е-маилот.

#### **Автентикација на корисници**

* Регистрираните корисници по верификација на својоата е-маил адреса може да се најават.
* Лозинките на корисниците се хешираат заради сигурност.

#### **Заборавена лозинка**

* Корисниците кои ја заборавиле својата лозинка може да испратат барање за ресетирање на лозинката.
* На е-маилот на корисникот пристигнува линк за ресетирање на лозинката за повторно да можат успешно да се најават.

### **Управување со проекти**

#### **Создавање и ажурирање на проекти**

* Корисниците може да создаваат нови проекти за ентериерен дизај.
* Интерфејсот овозможува додавање, бришење и преместување на 3D објекти во зададената 3D соба со помош на „drag and drop“ функционалноста и со користење на одредени копчиња од тастатурата.
* Можност за зачувување, ажурирање или бришење на проекти.

#### **Преглед на проекти**

* Преглед на сопствените создадени проекти во табела.
* Функционалност за сортирање по повеќе карактеристики на проектите, со што се подобрува управувањето на проектите.

### **3D модели**

#### **Поставување на 3D модели**

* Администраторите може да постават 3D модели, кои се достапни за сите корисници.
* Обичните корисници може да поставуваат 3D модели само за лична употреба.

#### **Ажурирање и бришење на 3D модели**

* Корисниците може да ги ажурираат податоците на сопствените 3D модели
* Корисниците исто така имаат можност и за бришење на сопствените 3D модели

#### **Преглед на 3D модели**

* Преглед на сопствените 3D модели
* Функционалност за сортирање по повеќе карактеристики на 3D моделите, со што се подобрува управувањето на 3D моделите.

### **Сигурност и автентикација**

* Најавените корисници имаат можност да ја променат својата лозинка со потврдување на моменталната.
* Функционалност за одјавување за да обезбеди сигурност на сметката на корисникот.

# **Технологии и алатки**

### **Фронтенд технологи**

* **React.js:** За изградба на фронтендот, се користи React.js. React.js е популарна JavaScript библиотека за изградба на интерактивни и сигурни кориснички интерфејси.
* **Three.js:** Three.js се користи за 3D компонентите на апликацијата, овозможувајќи создавање и манипулација на 3D објекти и сцени.
* **Material UI:** Фронтендот на RoomRevamp е изграден користејќи Material UI заради неговите веќе готови и прилагодливи компоненти за кориснички интерфејс, што го олеснува процесот на развој.

### 

### **Бекенд технологии**

* **PHP (Laravel 10):** Бекендот на RoomRevamp се базира на PHP, специфично Laravel 10. Laravel обезбедува силна и ефикасна рамка за изградба на веб апликации со фокус на лесна синтакса и функционалности за развивачите.

### **Меил сервис**

* **Mailtrap.io:** Mailtrap.io се користи како провајдер за меил сервисот за управување на комуникацијата преку е-маил, како верификација на е-маил и ресетирање на лозинката.

### **База на податоци**

* **PostgreSQL:** PostgreSQL служи како систем за управување на релациони база на податоци (RDBMS), обезбедувајќи складирање и пребарување на податоци на структуриран и сигурен начин.

### **Контрола на верзии и хостинг**

* **GitHub:** Изворниот код на RoomRevamp се хостира на GitHub, обезбедувајќи контрола на верзии и можност за соработка при развојот.

### **Алатки за развој**

* **Visual Studio Code (VS Code):** За развојот на апликацијата, ја користев алатката Visual Studio Code (VS Code), која служи како интегрирана развојна средина (IDE).
* **Оперативен Систем:** Апликацијата ја развивав на оперативниот систем "macOS".
* **Веб Прелистувач:** Апликацијата ја тестирав на веб прелистувачот "Google Chrome".

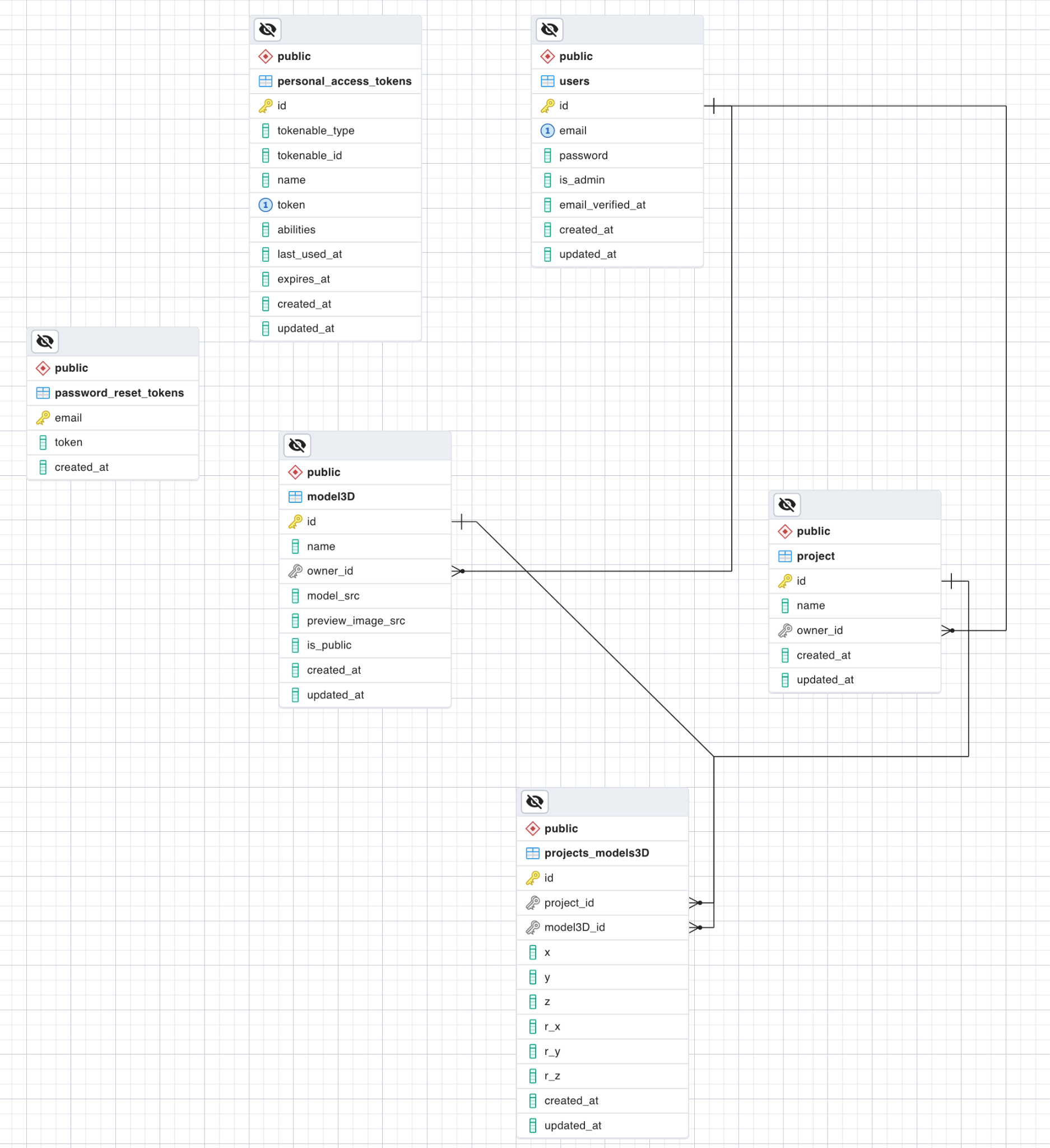
### **Континуирана интеграција и испорака (CI/CD) цевковод**

* **GitHub Webhooks:** GitHub Webhook-от е конфигуриран за да го известува Jenkins-от кога се прави „push“ на "main" гранката на репозиториумот.
* **Jenkins:** Jenkins се користи како сервер за автоматизација на CI/CD цевководот. Тој ги оркестира и автоматизира разните фази на процесот на доставување.
* **Kubernetes:** Kubernetes го оркестира доставувањето, скалирањето и управувањето на контејнерите, обезбедувајќи ефикасно функционирање на апликацијата.

#### **Фази на CI/CD цевководот:**

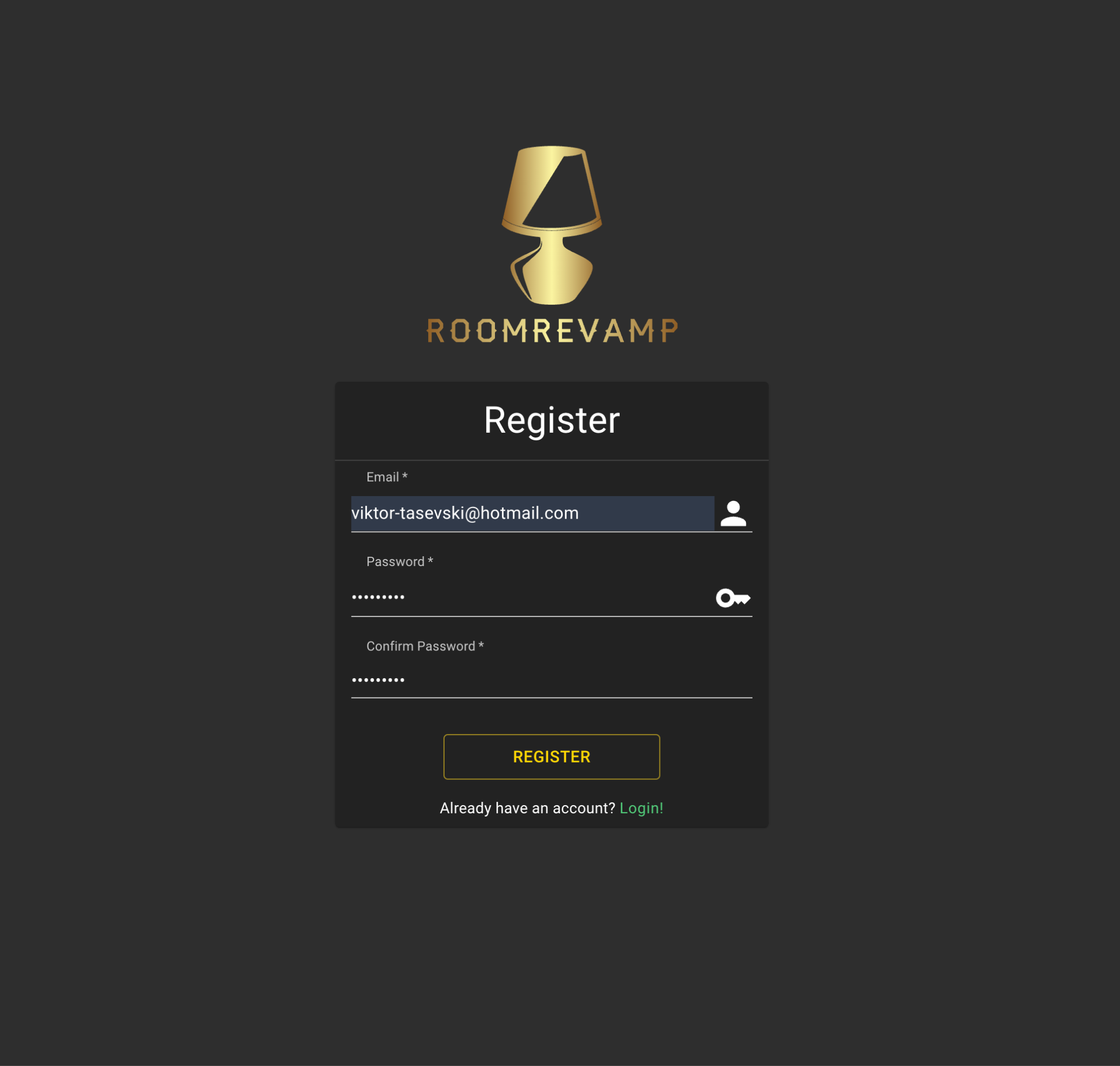
1. **Build and push backend:** CI/CD цевководот започнува со градење на бекенд кодот и создавање на Docker слика, која потоа се прикачува на DockerHub. Оваа фаза обезбедува дека најновиот бекенд код е контејнеризиран за „deployment“.
2. **Build and push frontend:** Се изградува кодот за фронтенд, и соодветна Docker слика се креира. Истата се прикачува на DockerHub, обезбедувајќи дека најновиот код за фронтенд е достапен за „deployment“.
3. **Deploying to kubernetes:** Во последната фаза, правиме нов „deployment“ на фронтендот и бекендот на локалниот Kubernetes кластер.

# **Функционалности**



*Слика-1. Модел на базата на податоци*

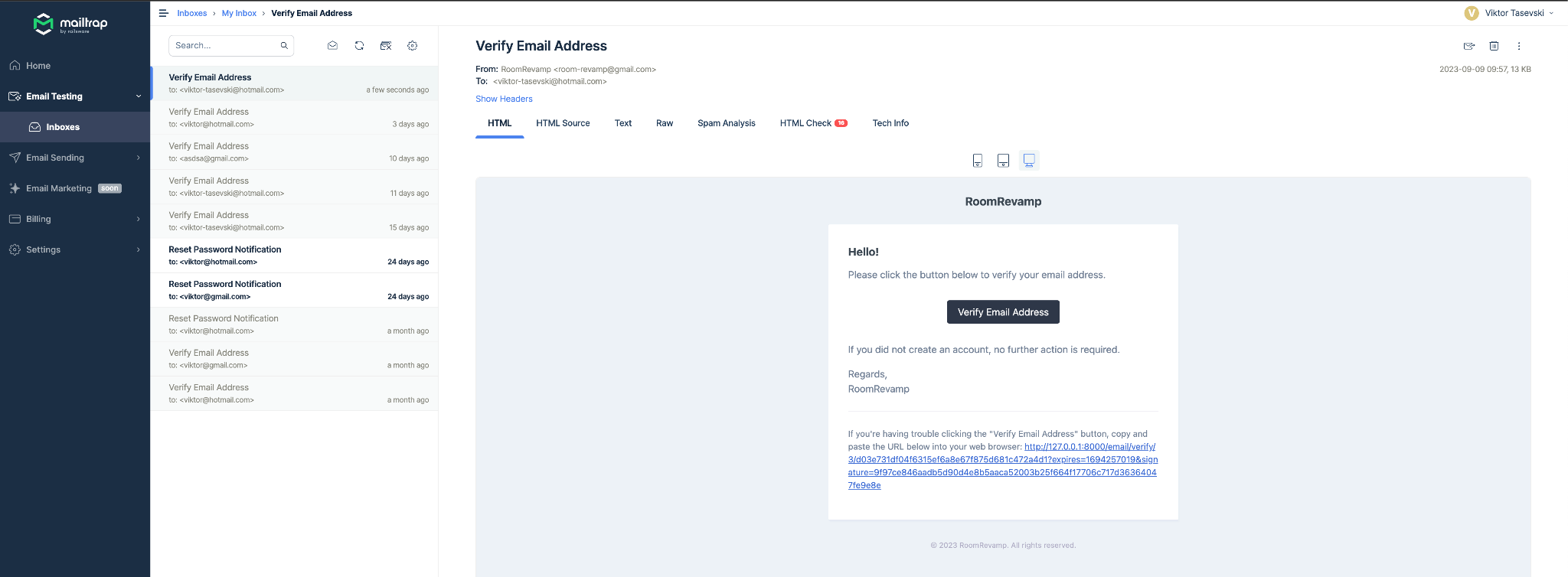
### **Регистрација на корисник**



*Слика-2. Страна за регистрација на корисници*

Во формата на Слика-2 корисникот треба да го внесе својот е-маил, посакуваната лозинка и истата да ја повтори. Доколку е-маилот е празна низа од карактери или не е валиден, под е-маил полето се прикажува соодветната грешка. Доколку лозинката е празна низа од карактери, се прикажува грешка испод полето. Доколк лозинката и потврди лозинка полињата не се еднакви тогаш се прикажува грешка испод потврди лозинка полето. Доколку е-маилот е зафатен тогаш се прикажува грешка испод потврди лозинка полето. Валидирањето на формата се извршува на клиентска форма, и доколку формата не ги задоволвуа поставените барања, не се испраќа барање до бекендот.

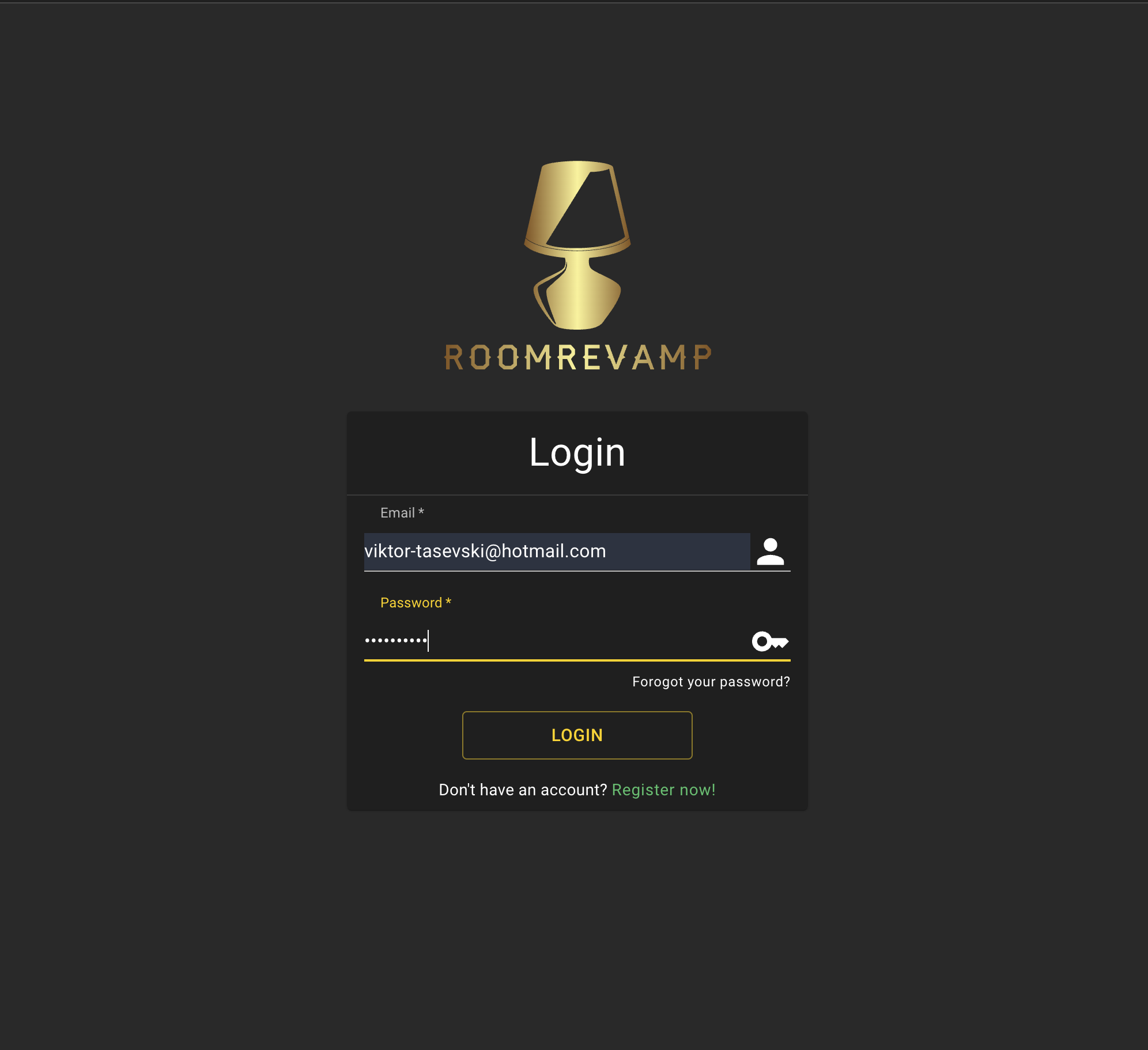
По притискање на копчето „Register“ доколку формата е валидна и барањето до бекендот е успешно, се испраќа е-маил до е-маилот на корисникот за да го потврди истиот и е пренасочен кон страната за најавување.



*Слика-3. Е-маил со линк за верификација на новорегистрирани корисници*

На Слика-3 можеме да видиме дека пристигнат е е-маилот за потврдување на е-маилот на корисникот. По притискање на копчето „Verify Email Address“, пренасочени сме кон бекенд страната за верификација на е-маил, на endpoint-от „/email/verify/{id}/{hash}“. По успешна верификација, бекендот прави пренасочување кон фронтендот на страната за најавување.

### **Најава на корисник**



*Слика-4. Страна за најавa на корисници*

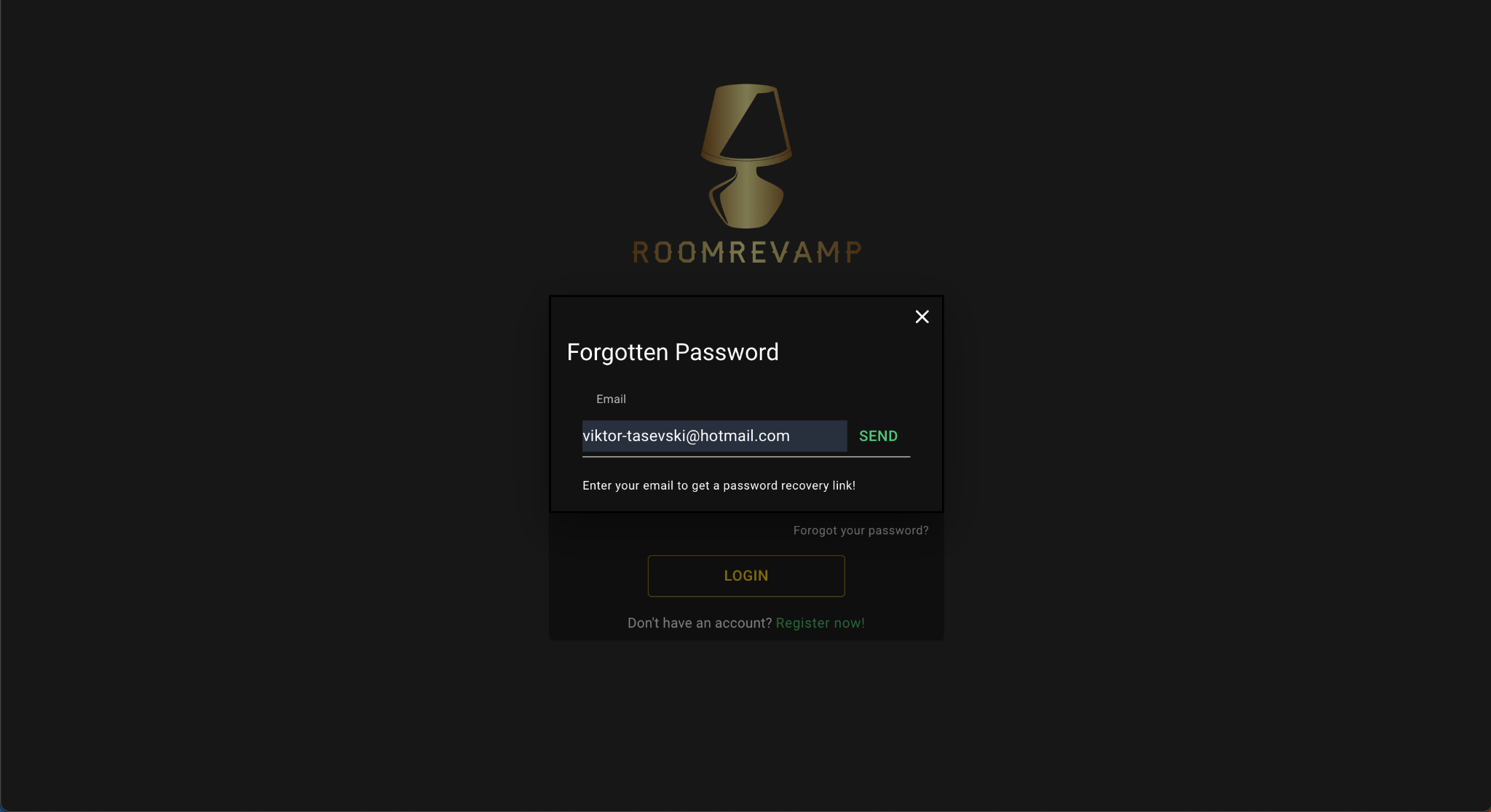
Во формата на Слика-4, корисникот потребно е да го пополни полето за е-маил со валиден е-маил и исто така полето за лозинка мора да е пополнето. Доколку не е исполент некој од условите за тоа што претставува валидна форма, се прикажува грешката под соодветното поле. Откако формата ќе има валиден статус со притискање на копчето „Login“ се испраќа барање до бекендот, ако барањето не е успешно грешката се прикажува под полето за лозинка, но ако барањето е успешно, го сетираме JWT-то во local storage и го пренасочуваме корисникот кон почетната страна.



*Слика-5. Автентикација и авторизација на веќе најавен корисник*

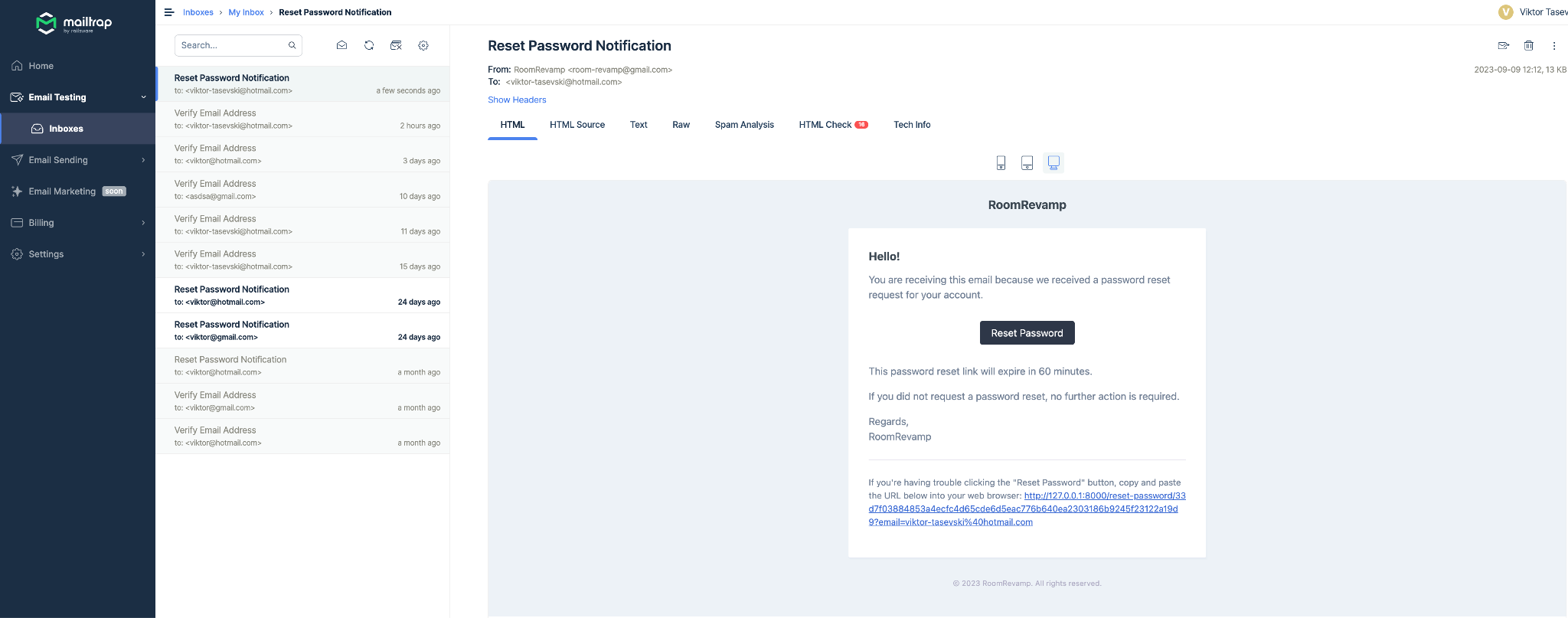
На Слика-5 го имаме кодот за тоа како би извршиле автентикација и авторизација на корисник кој веќе се има најавено и добиено JWT. Прво го зимаме JWT-то од local storage, доколку не постои корисникот е пренасочен на страната за најавување ( пренасочувањето не можеме да го видиме тука но на друго место во апликацијата имаме поставено услов доколку корисникот не постои и не се вчитува тогаш корисникот да биде пренасочен кон страната за најавување ), доколку постои тогас со функцијата „setAuthHeader“ на axiost инстанцата JWT-то го поставуваме како header за авторизација кој би се испраќал со секое барање. Потоа на ред 18 се обидуваме да ги земеме податоците за моменталиот кориснико, доколку е неуспешно header-от на axios инстанцата го сетираме на празна низа од карактери и корисникот ќе биде пренасочен на страната за најавување, доколку барањето е успешно, податоците за корисникот од бекендот се запамтување и корисникот е пренасочен на почетната страна.

### **Заборавена лозинка**



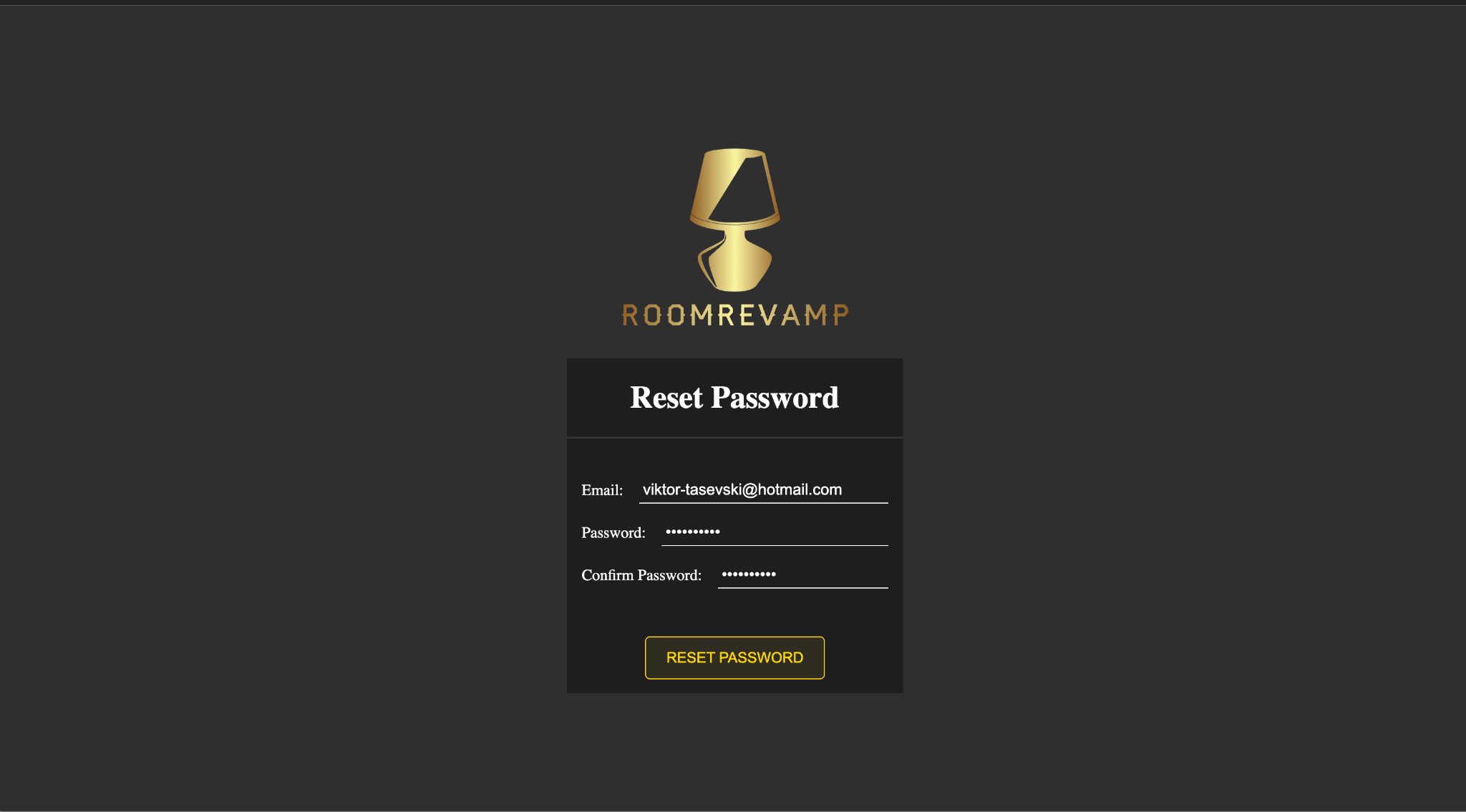
*Слика-6. Форма за барање за заборавен лозинка*

На страната за најавување под полето за лозинка имаме копче „Forgot your password?“, со негово притискање ни се отвара модал со форма за барање за ресетирање на лозинката. Потребно е корисникот да внесе валиден е-маил и со притискање на копчето „SEND“ се испраќа е-маил до внесениот е-маил со линк за ресетирање на лозинката. По успешното барање модалот се затвара.



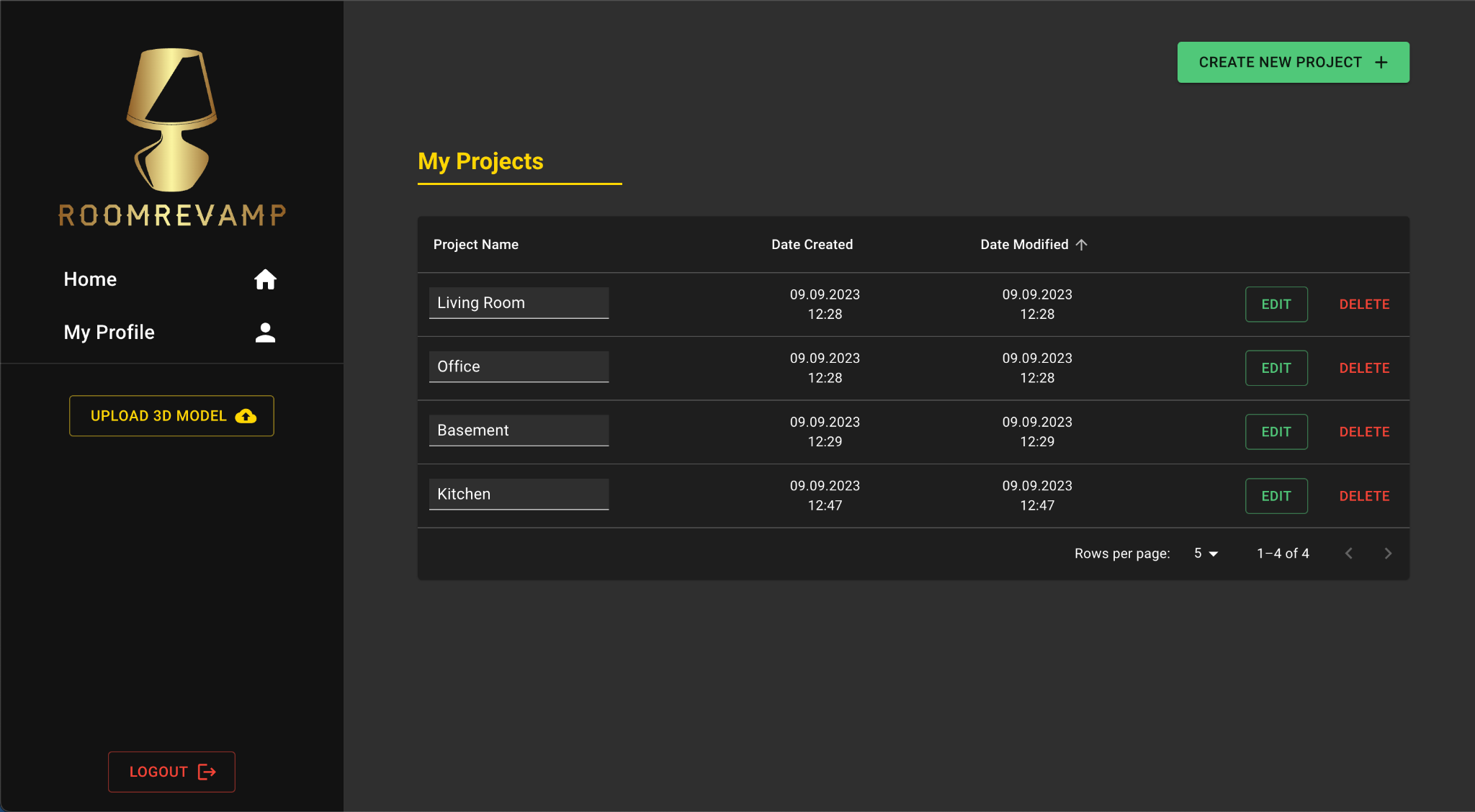
*Слика-7. Е-маил со линк за ресетирање на лозинката*

На Слика-7 можеме да го видиме е-маилот со линкот за ресетирање на лозинката. Со притискање на копчето „Reset Password“ или линкот најдоле пренасочени сме кон бекенд страната за ресетирање на лозинката.



*Слика-8. Форма за ресетирање на лозинката*

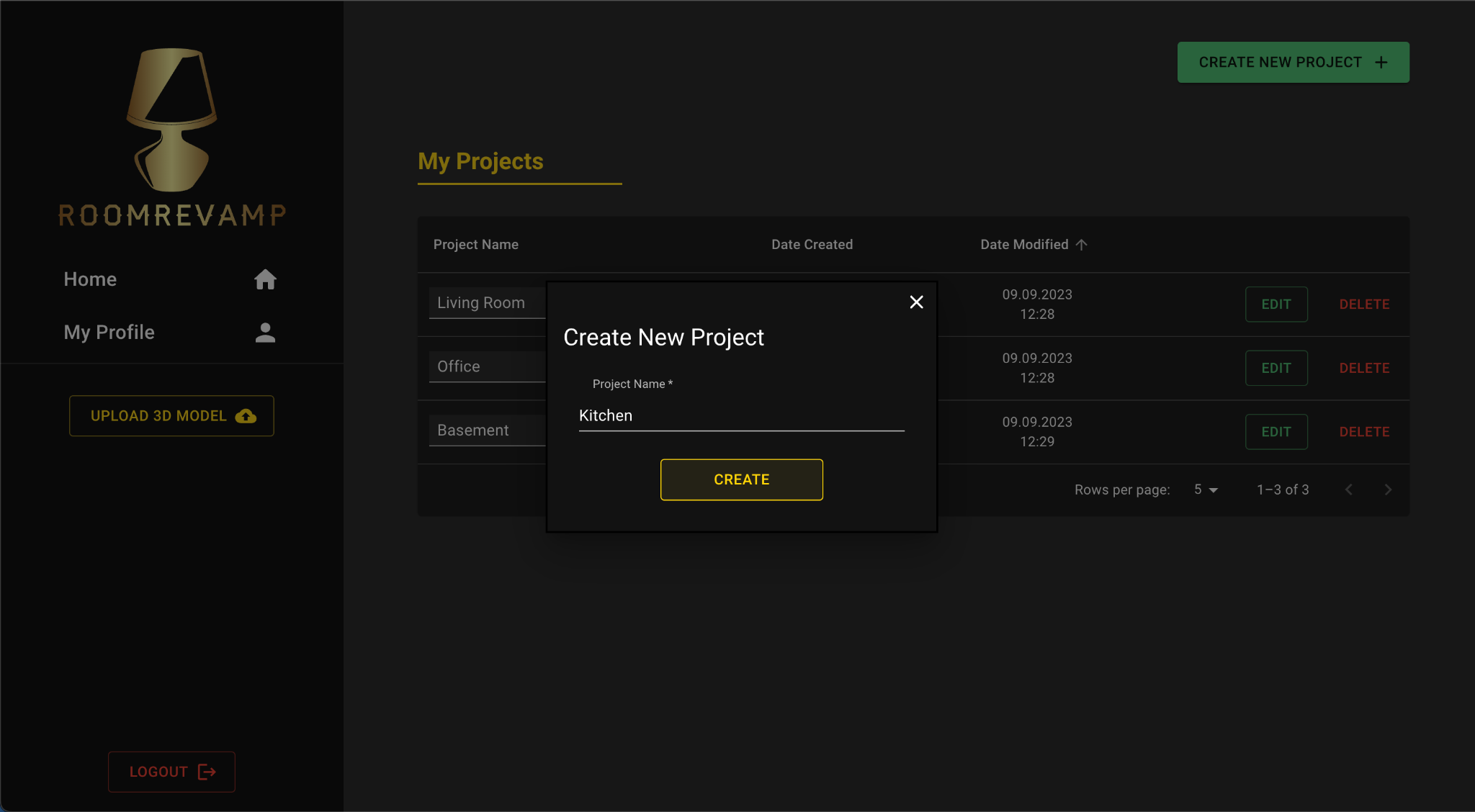
На Слика-8 потребно е да ги пополниме сите полиња на формата за ресетирање на лозинката. По притискање на копчето „RESET PASSWORD“, пренасочени сме кон фронтендот на страната за најавување.

*Слика-9. Почетна страна*

На Слика-9 имаме преглед на почетната страна. Во наредните страни ќе бидат објаснети сите функционалности.

### 

### **Создавање на нов проект**

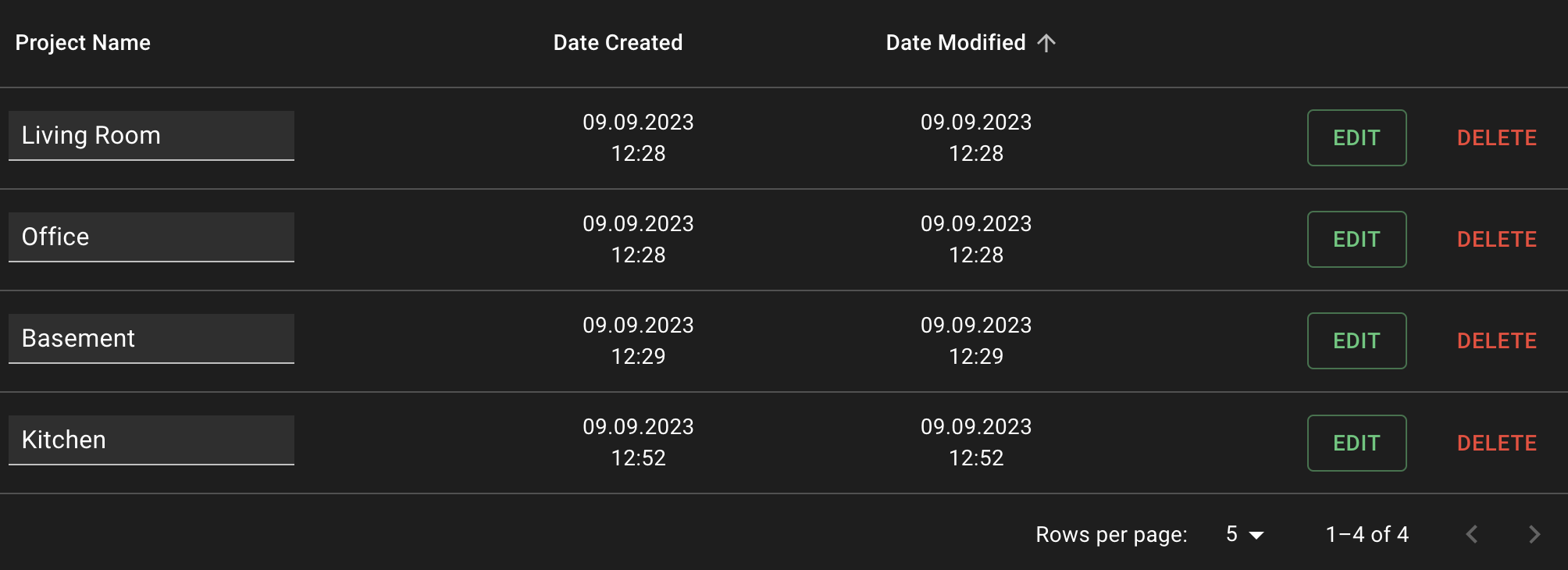


*Слика-10. Форма за создавање нов проект*

Со притискање на копчето „CREATE NEW PROJECT“ во десниот горен агол, ни се отвара модал со форма за создавање на нов проект. Тука потребно е само да го внесеме името на нашиот проект и да притиснеме на копчето „CREATE“. По притискање на копчето „CREATE“ испраќаме барање за создавање на нов проект до бекендот и тој проект е додаден во табелата на сопствени проекти и модалот се затвара.

### 

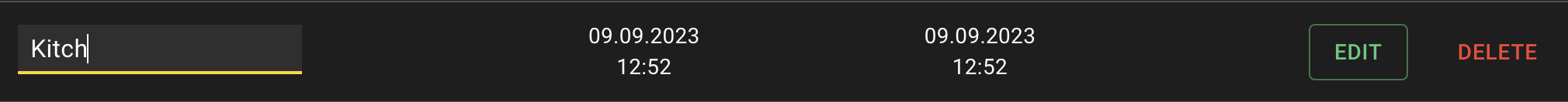
### **Преглед на проекти**



*Слика-11. Табела на создадени проекти*

На Слика-11 имаме преглед врз нашите проекти, опишани со името, датумот на создавање и датумот на последно ажурирање. Исто така и копчиња за ажурирање и бришење.

### **Ажурирање и бришење на проект**

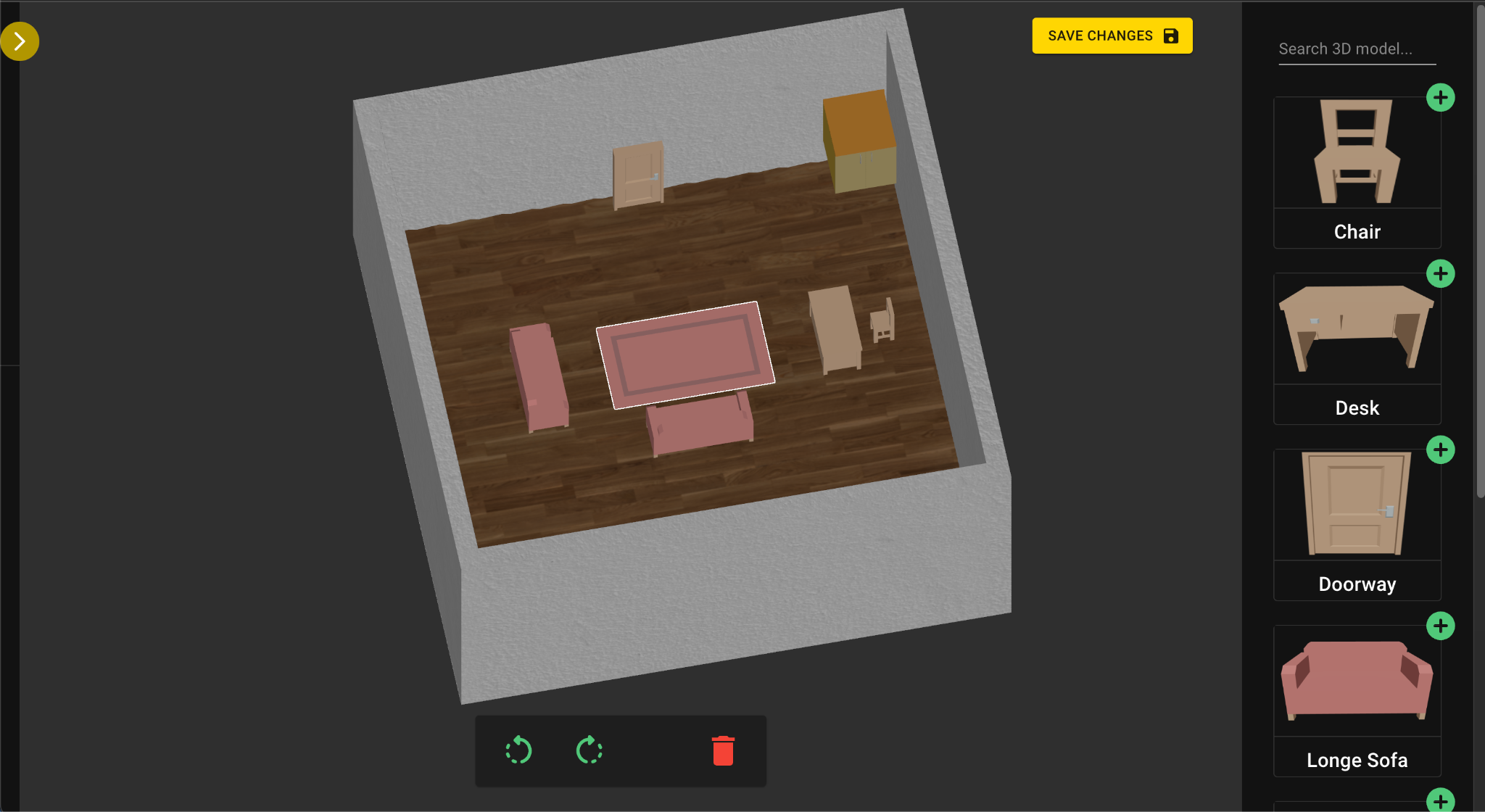


*Слика-12. Еден ред од табелата на создадени проекти*

Името на проектот можеме директно да го ажурираме во табела, може да видиме дека полето е editable. Со губење на фокусот на полето за име на проектот, се испраќа барање до бекендот за ажурирање на името. Со притискање на копќето „EDIT“ пренасочени сме кон страната за 3D дизајнирање на ентериерот. Со притискање на копчето „DELETE“ се испраќа барање до бекендот за бришење на проектот, доколку барањето е успешно проектот е тргнат од табелата на сопствени проекти.

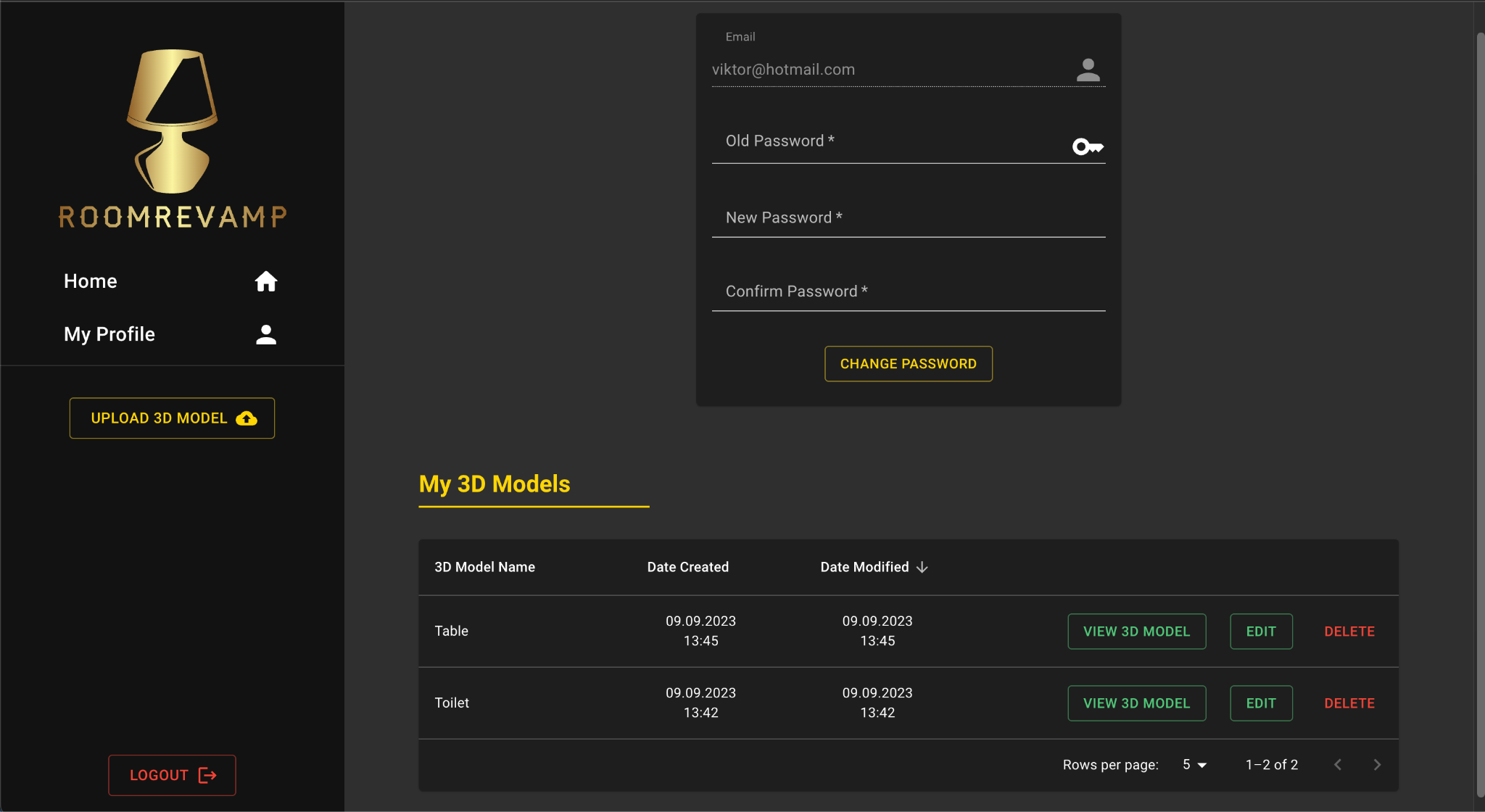
### 

### **Уредување на ентериерот**



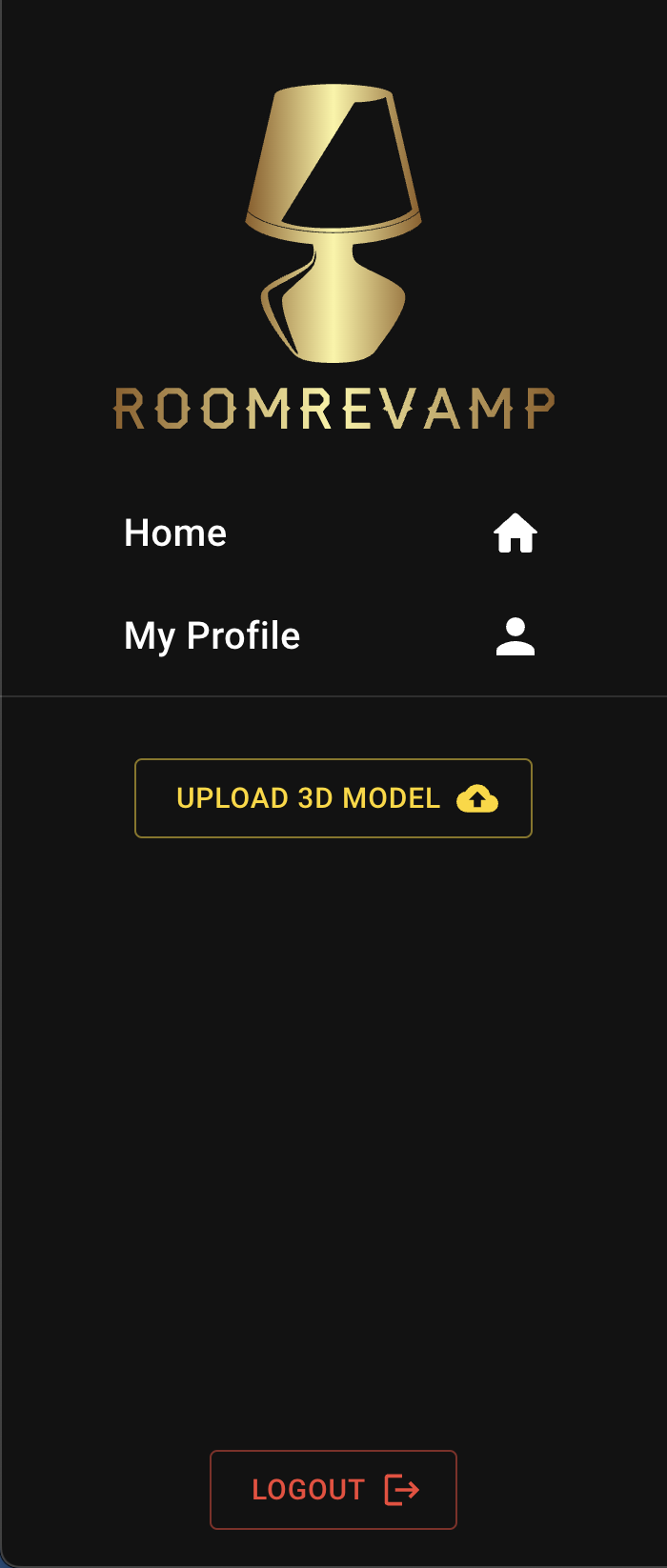
*Слика-13. Страна за уредување на етериерот*

На Слика-13 е страната за 3D уредување. Во левиот горен агол се наоѓа копче со стрелка кое го оствара страничното мени прикажано на Слика-15. Десно на екранот се наоѓа листа од 3D модели кои можеме да ги вметнеме во ентериерот со притискање на соодветното копче зелено копче со плус на некоја од картичките во листата. Исто така имаме филтрирање според името на 3D моделот. Во средината на екранот можеме да ја видиме 3D собата со наместени 3D елементи. Со „drag-and-drop“ можеме да ги распоредиме елементите и исто така со стрелките на тастатурата. Со мрдање на глувчето и притискање на некоја од стрелките, елементите можеме да ги мрдаме напред, назад, лево и десно, но за да можеме да ги мрдаме горе и доле мораме да го притиснеме копчето „y“, и по притискање наместо моделот да се движи по z-оската сега ќе се движи по y-оската во просторот. Со селектирање на одреден 3D елемент, во централниот долен дел на екранот се прикажуваат одредени контроли кои можеме да ги извршуваме врз селектираниот елемент, и тоа ротирање на лево за 90 степени, ротирање на десно за 90 степени и бришење на елементот од собата. Во горниот подесен дел на екранот го имаме копчето „SAVE CHANGES“ кое иницијално е оневозможено, но чим се детектира измена во собата копчето е достапно да се претисне. Со претискање на истото се испраќаат направените измени до бекендот и потоа копчето повторно е оневозможено.



*Слика-14. Страна „Мој Профил“*

На Слика-14 имаме преглед на „My Profile“ страната. Во наредните страни ќе бидат објаснети сите функционалности.

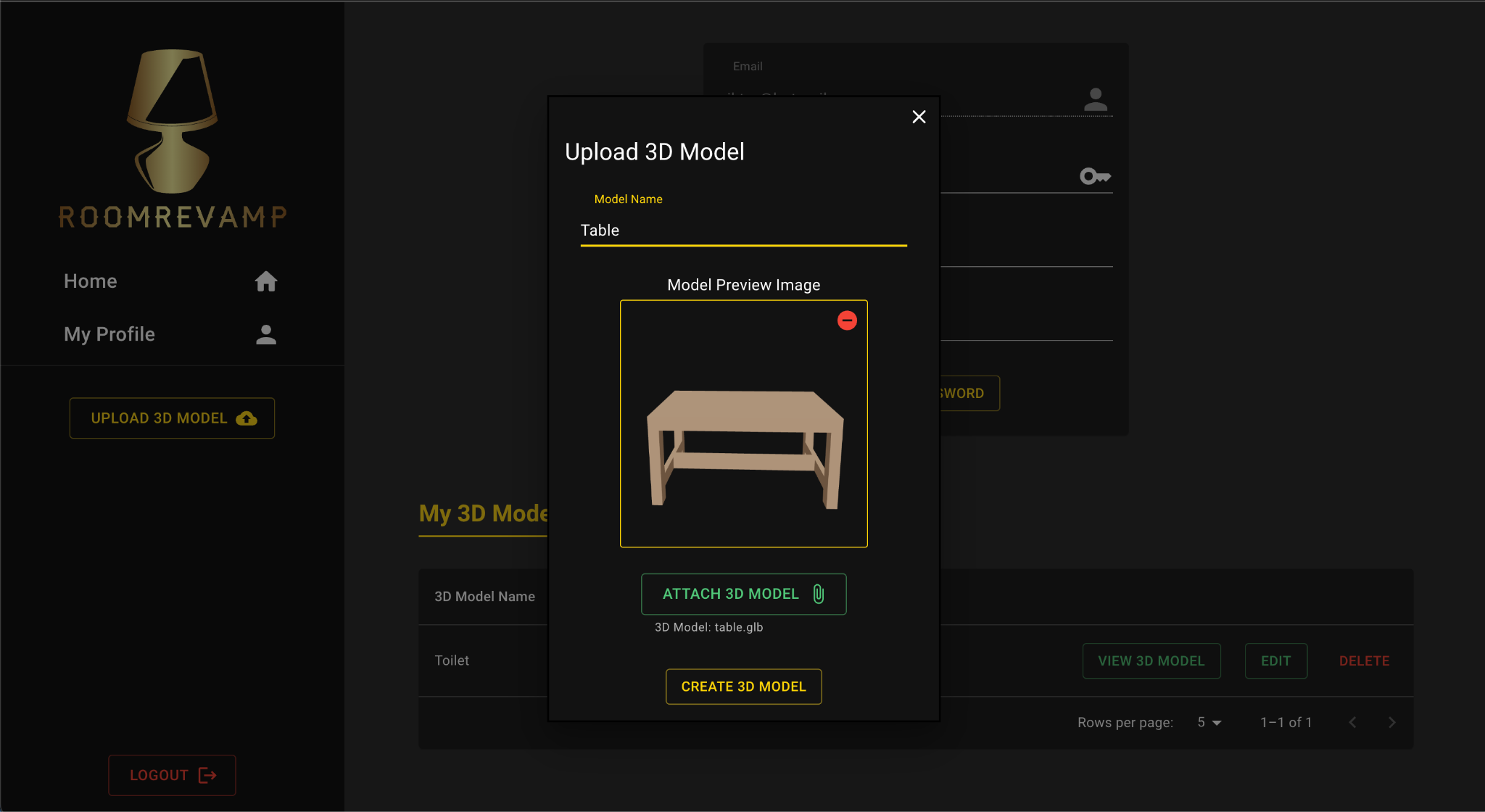


*Слика-15. Странично мени*

На Слика-15 го имаме страничното мени на апликацијата. Прво поставено е логото на RoomRevamp, потоа навигацијата низ апликацијата, па копче да поставување на 3D модел и најдоле го имаме копчето „LOGOUT“ за одјавување од апликацијата.

### 

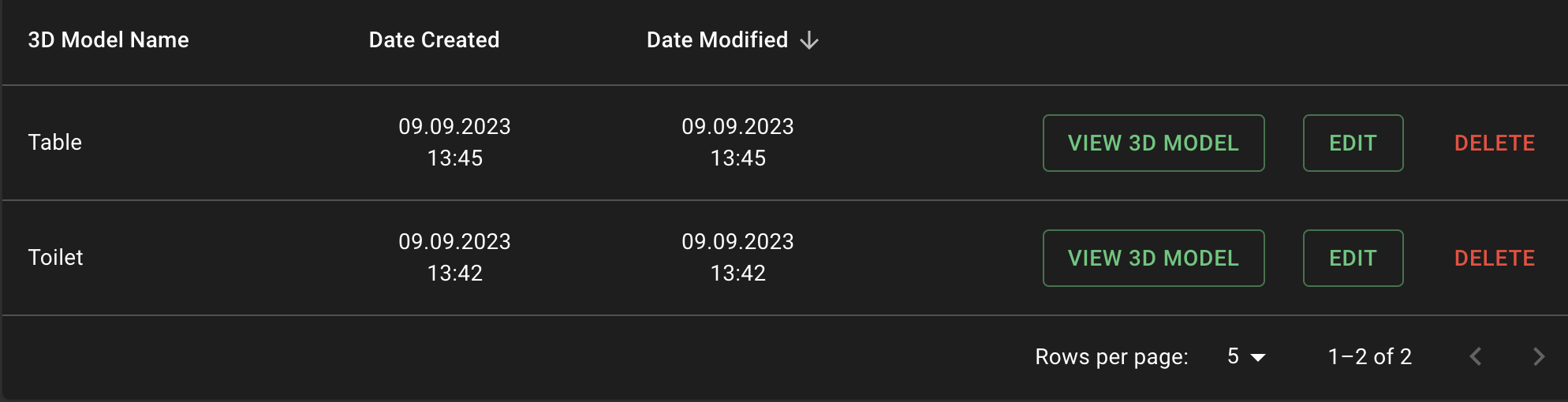
### **Додавање на 3D модел**



*Слика-16. Додавање на 3D модел*

Со притискање на копчето „UPLOAD 3D MODEL“ од страничното мени се отвара модал со форма за поставување на 3D модел. Формата е сочинета од полиња за име, прегледна слика од моделот, 3D датотека (.obj), и chekbox за тоа дали моделот ќе биде споделен со останатите корисници или не (ова поле е видливо само за Админите). Сите полиња мора да бидат пополнети за формата да биде валидна и да се овозможи испраќање на барање на бекендот. По притискање на копчето „CREATE 3D MODEL“ се испраќа барање за поставување на 3D моделот. По успешно поставување, 3D моделот се додава во табелата на свои поставени 3D модели и модалот се затвара.

### **Преглед на 3D модели**



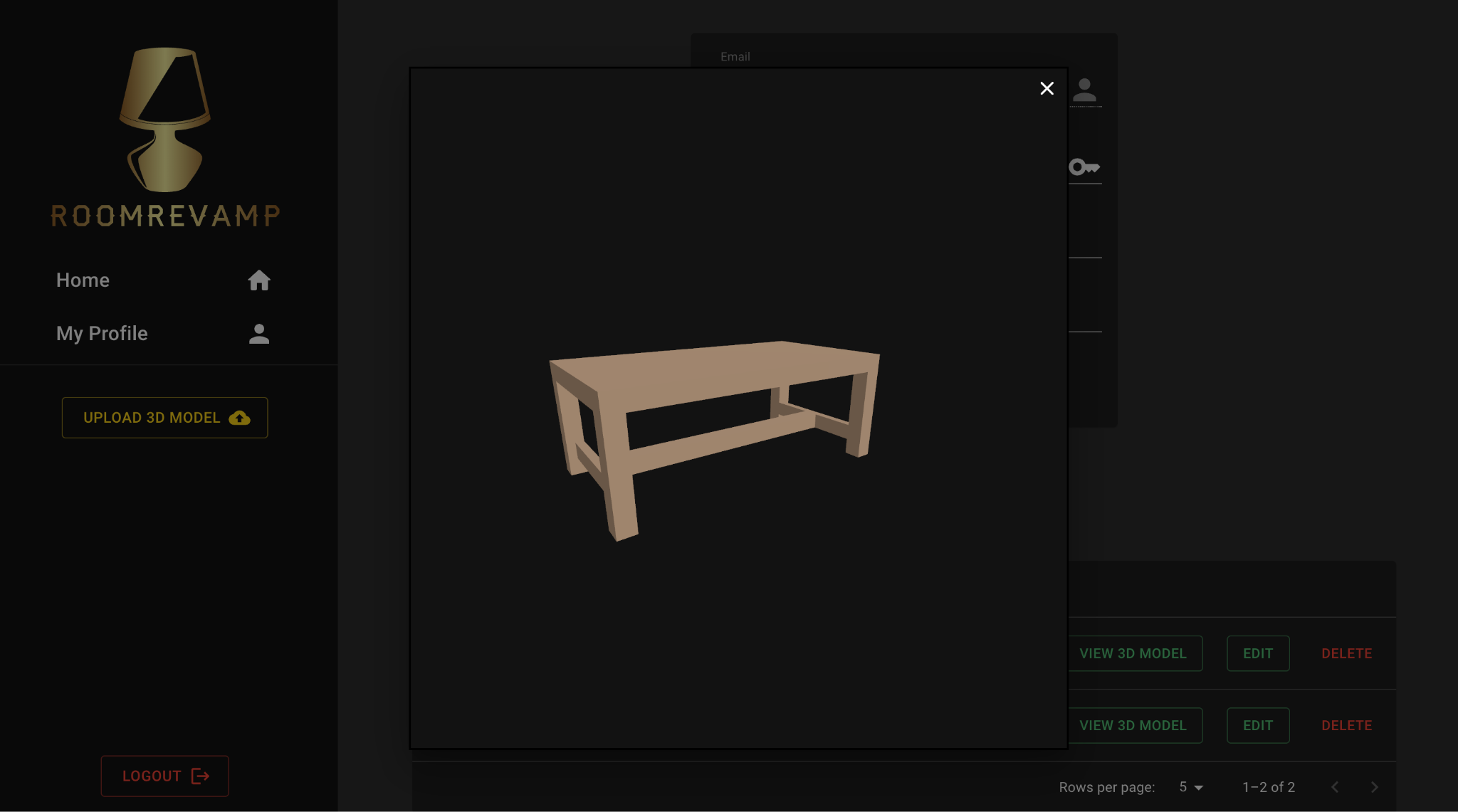
*Слика-17. Табела на 3D модели ( Обичен Корисник )*

На Слика-17 имаме преглед врз нашите поставени 3D модели, опишани со името, датумот на создавање и датумот на последно ажурирање. Исто така и копчиња за 3D преглед на моделот, ажурирање и бришење.



*Слика-18. Табела на 3D модели ( Админ )*

Слика-18 е иста како Слика-17 со додаток на колоната „Is Public“ која е видлива само за Админите на апликацијата. Доколку вредноста на оваа колона е „Yes“ тогаш 3D моделот е споделен со сите корисници на апликацијата, доколку е „No“ тогаш само Админот кој што го поставил може да го користи.

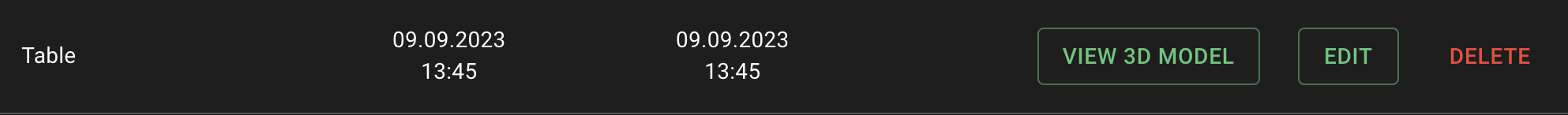


*Слика-19. 3D приказ на избраниот 3D модел*

По притискање на копчето „VIEW 3D MODEL“ од некој ред од табелата на свои поставени 3D модели, се отвара модал кој што го прикажува соодветниот 3D модел.

### 

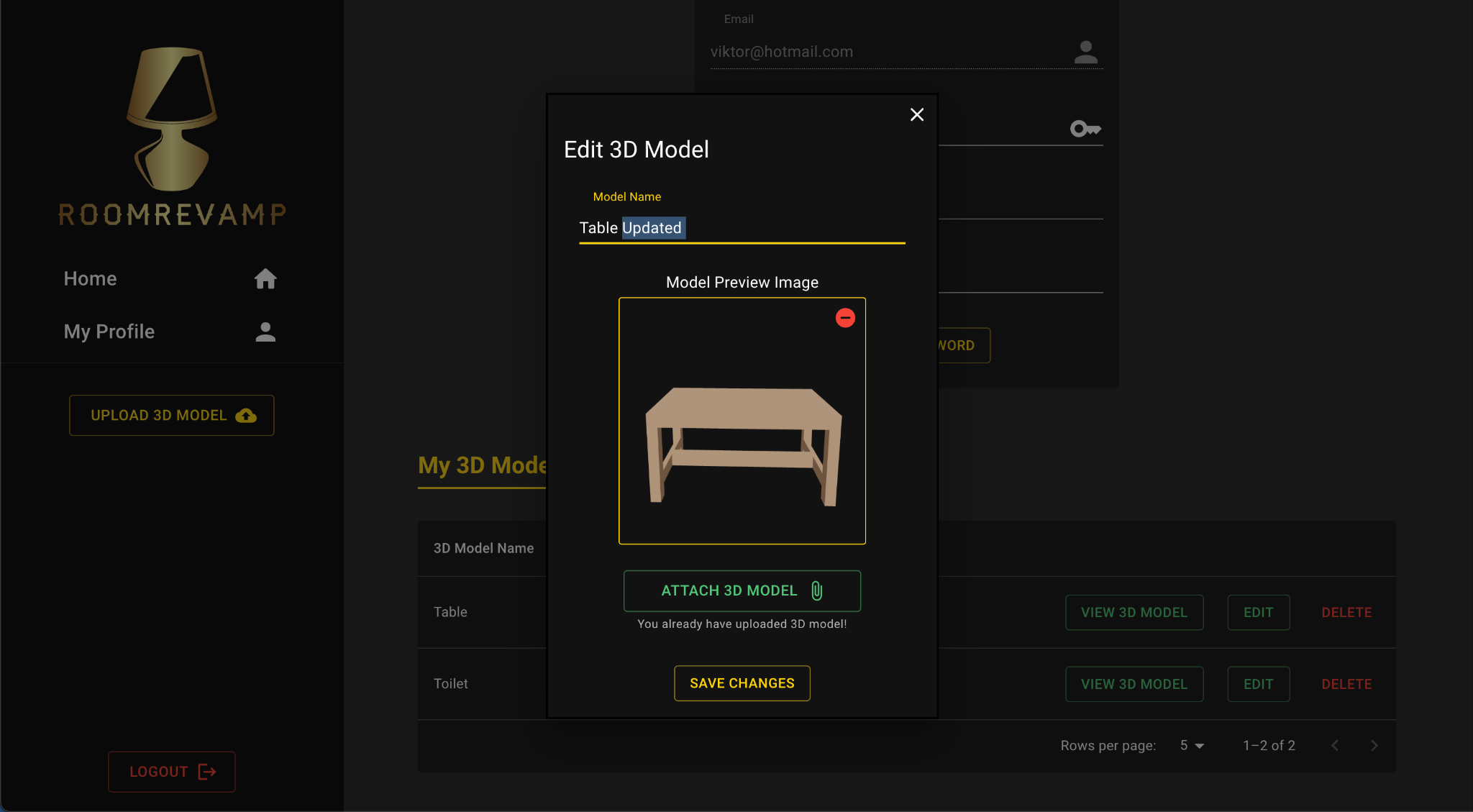
### **Ажурирање и бришење на 3D модел**



*Слика-20. Еден ред од табелата на 3D модели ( Обичен Корисник )*

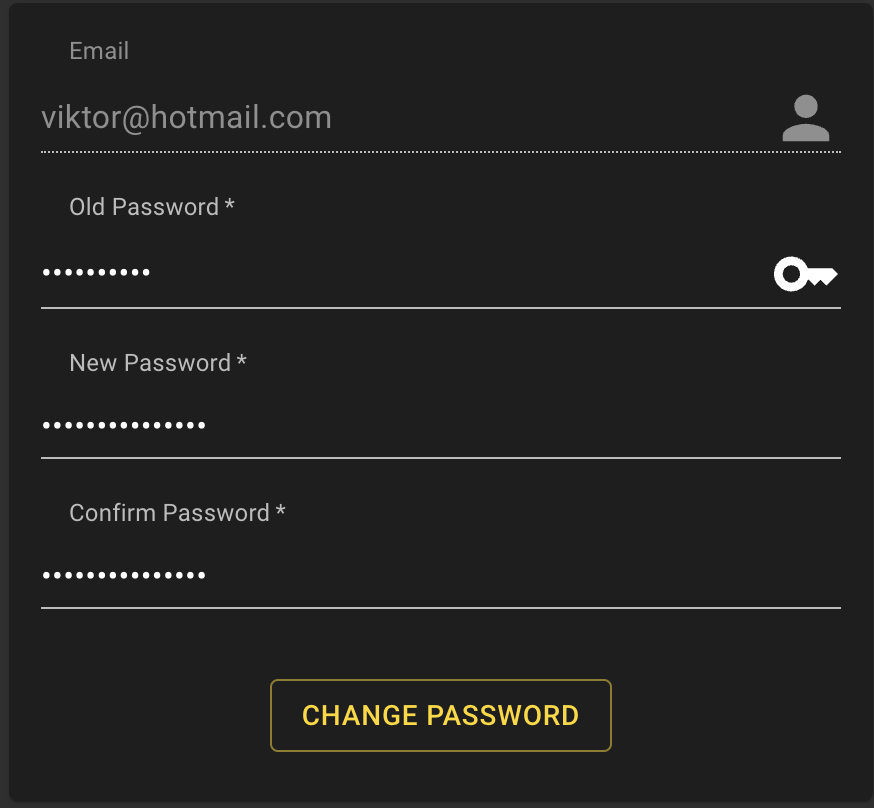
Со притискање на копчето „EDIT“ ни се прикажува модалот на Слика-21. Оваа форма е иста како и формата за поставување на 3D модел. По потреба или желба ги изменуваме посакуваните податоци за 3D моделот и со притискање на копчето „SAVE CHANGES“ се испраќа барање до бекендот за ажурирање на 3D моделот. По успешно ажурирање модалот се затвара.

Со притискање на копчето „DELETE“ се испраќа барање до бекендот за бришење на 3D моделот. По успешно бришење моделот се остранува од табелата на свои поставени 3D модели.



*Слика-21. Ажурирање на 3D модел*

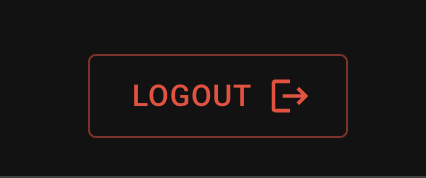
### **Промена на лозинка**



*Слика-22. Форма за промена на лозинка*

Во средината на екранот на страната „My Profile“ ја имаме формата за промента на лозинката на најавениот корисник. Полето за е-маил е оневозможено и служи само како информација за тоа кој корисник е најавен. За да ја промениме моменталната лозинка потребно е нејзе да ја внесеме па потоа новата и за крај да ја потврдиме новата лозинка. За формата да е валидна за да се дозволи испраќање на барање до бекендот, потребно е сите полиња не да бидат празни и новата лозинка и лозинката за потврда да се еднакви. Со притискање на копчето „CHANGE PASSWORD“ се испраќа барање до бекендот за промена на лозинката. На бекендот се проверува дали вредноста на „old password“ полето од формата е еднаква на моменталната лозинка на корисникот, доколку е еднаква тогаш вредноста на полето „new password“ се поставува како нова лозинка (хешираната вредност). По успешна промена на лозинката полиња на формата на фронтендот се поставуваат на празна низа од карактери.

### **Одјававување**



*Слика-23. Копче за одјава*

Со притискање на копчето за одјавување од страничното мени, се испраќа барање до бекендот за поништување на JWT-то кој бил создаден при најавување, потоа JWT-то е отстранет од local storage и корисникот е пренасочен кон страната за најавување.

# 

# **Стартување на апликацијата**

### **Начин 1**

Потребно е да имате инсталирано yarn ( <https://classic.yarnpkg.com/lang/en/docs/install> ),

php 8 ( <https://www.php.net/downloads.php> ),

composer ( <https://getcomposer.org/download/> ).

Потребно е да имате подигнато postgresql сервер.

#### **Стартување на бекенд**

Прво во root директориумот на бекендот во датотеката .env првоверете ги environment variables-от дали се согласуваат со вашите барања.

За да ги инсталираме потребните пакети ја извршуваме командата „***composer install***“ во root-от на проектот.

За да ги извршиме миграциите ја извршуваме командата „***php artisan migrate***“.

За да го стартуваме серверот потребно е да ја извршиме командата „***php artisan serve***“.

#### **Стартување на фронтенд**

Прво во root директориумот на фронтендот во датотеката .env првоверете ги environment variables-от дали се согласуваат со вашите барања.

За да ги инсталираме потребните пакети ја извршуваме командата „***yarn install***“ во root-от на проектот.

Доколку сакате да го стартувате production серверот, тогаш потребно е да ја извршите командата „yarn build“, потоа „**yarn global add serve**“ и на крај „**serve -s build**“.

Доколку сакате да го стартувате development серверот, тогаш извршете ја само командата: „**yarn start**“

### **Начин 2**

Најлесно е да инсталирате docker desktop ( [macOs](https://docs.docker.com/desktop/install/mac-install/), [windows](https://docs.docker.com/desktop/install/windows-install/), [linux](https://docs.docker.com/desktop/install/linux-install/) ), бидејќи ги содржи сите потребни алатки.

#### **Стартување на бекенд, фронтенд и база**

Отворете ја датотека „docker-compose.yml“ која се наоѓа на „src/devops/docker-compose.yml“ и проверете ги environment variables-от како и volume bindings-от дали се согласуваат со вашите барања.

Навигирајте се во директориумот „devops“ ( src/devops ) и стартувајте ги сите сервиси со командата „***docker compose up***“.

### **Начин 3**

Најлесно е да инсталирате docker desktop ( [macOs](https://docs.docker.com/desktop/install/mac-install/), [windows](https://docs.docker.com/desktop/install/windows-install/), [linux](https://docs.docker.com/desktop/install/linux-install/) ), бидејќи содржи голем дел од потребните алатки.

Потребно е да имате инсталирано k3d ( <https://k3d.io/v5.6.0/#installation> ),

kubectl ( [macOs](https://kubernetes.io/docs/tasks/tools/install-kubectl-macos/), [windows](https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/), [linux](https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/) ).

Прво навигирајте се во директориумот „src/devops“.

Треба да имаме создадено и сопствен kubernetes cluster и истиот можеме да го создадеме со извршување на скриптата „create-cluster.sh“ со командата „***sh create-cluster.sh***“.

#### **Стартување на база**

За да го стартуваме postgresql серверот, потребно е да ја извршиме скриптата „postgres-deploy.sh“ со командата „***sh postgres-deploy.sh***“.

#### **Стартување на бекендот**

Откако ќе се подигне postgresql серверот, ќе го подигнеме ибекенд серверот со извршување на скриптата „deploy-backend.sh“ со командата „***sh backend-deploy.sh***“.

По подигањето на бекенд серверот потребно е да го извршиме миграциите мануелно. Па така со извршување на командата „***kubectl get pods***“ во терминалот ќе ги добиеме сите подови кои се „живи“. Нас не интересира еден од бекенд подовите чии име би било „backend-deployment-{podId}“. Потребно е да влеземе во еден од подовите, и тоа го правиме со извршување на командата „***kubectl exect -it {POD\_NAME} -- /bin/bash***“ во терминалот. Откако сме влегле внатре, со извршување на командата „***php artisan migrate***“ ги извршуваме и потребните миграции. Со командата „***exit***“ искачаме надвор од подот.

#### **Стартување на фронтенд**

За да го стартуваме фронтенд серверот, потребно е да ја извршиме скриптата „client-deploy.sh“ со командата „***sh client-deploy.sh***“.

#### **Влез во нашата апликација**

Откако сме ги стартувале серверите за базата, бекендот и фронтендот, потребно е и некако да можеме да ги пристапиме. Па затоа за крај потребно е и да го аплицираме и манифестот „ingress.yml“ со извршување на командата „kubectl apply -f ingress.yml“ во терминалот.

За да можете да го пристапите фронтендот на room-revamp.com, треба овој домен и доменот на бекендот да ги додадете во датотеката „/etc/hosts“, која служи за мапирање на домени со IP адреси за локалниот DNS ( macOs ). За windows погледнете го следниот линк <https://helpdeskgeek.com/networking/edit-hosts-file/>. 