



Универзитет „Св. Кирил и Методиј“ во Скопје  
**ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И  
КОМПЈУТЕРСКО ИНЖЕНЕРСТВО**

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

<b>Вовед.....</b>	<b>4</b>
<b>Спецификација на софтверски барања.....</b>	<b>5</b>
1. Управување со корисници.....	5
1.1. Рољи на корисници.....	5
1.2. Регистрација на корисници.....	5
1.3. Автентикација на корисници.....	5
1.4. Заборавена лозинка.....	6
2. Управување со проекти.....	6
2.1. Создавање и ажурирање на проекти.....	6
2.2. Преглед на проекти.....	6
3. 3D модели.....	6
3.1. Поставување на 3D модели.....	6
3.2. Ажурирање и бришење на 3D модели.....	6
3.3. Преглед на 3D модели.....	6
4. Сигурност и автентикација.....	7
<b>Технологии и алатки.....</b>	<b>7</b>
Фронтенд технологи.....	7
Бекенд технологии.....	7
Меил сервис.....	8
База на податоци.....	8
Контрола на верзии и хостинг.....	8
Алатки за развој.....	8
Континуирана интеграција и испорака (CI/CD) цевковод.....	8
Фази на CI/CD цевководот:.....	9
<b>Функционалности.....</b>	<b>10</b>
Регистрација на корисник.....	11
Најава на корисник.....	13
Заборавена лозинка.....	15
Создавање на нов проект.....	18
Преглед на проекти.....	19
Ажурирање и бришење на проект.....	19
Уредување на ентериерот.....	20
Додавање на 3D модел.....	23
Преглед на 3D модели.....	24
Ажурирање и бришење на 3D модел.....	25
Промена на лозинка.....	26
Одјавување.....	27
<b>Проблем со ThreeJs при rerender на апликацијата.....</b>	<b>27</b>
Обрзложение на проблемот.....	27

Совети за надминување на проблемот.....	28
<b>Стартување на апликацијата.....</b>	<b>29</b>
Начин 1.....	29
Стартување на бекенд.....	29
Стартување на фронтенд.....	29
Начин 2.....	30
Стартување на бекенд, фронтенд и база.....	30
Начин 3.....	30
Стартување на база.....	31
Стартување на бекендот.....	31
Стартување на фронтенд.....	31
Влез во нашата апликација.....	31

## 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 околии. 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.

Мојата 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 релационите 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 проект извештај, ќе се навлеземе подетално во методологијата која беше користена во процесот на развој и ќе ги истражime 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**

### **1. Управување со корисници**

#### **1.1. 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.

#### **1.2. 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.

#### **1.3. Автентикација на корисници**

- Регистрираните корисници по верификација на својата е-маил адреса може да се најават.

- Лозинките на корисниците се хешираат заради сигурност.

#### **1.4. 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.

## **2. Управување со проекти**

### **2.1. 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.

### **2.2. 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.

## **3. 3D модели**

### **3.1. 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.

### **3.2. 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

### **3.3. 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.

#### 4. 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 технологии

- **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

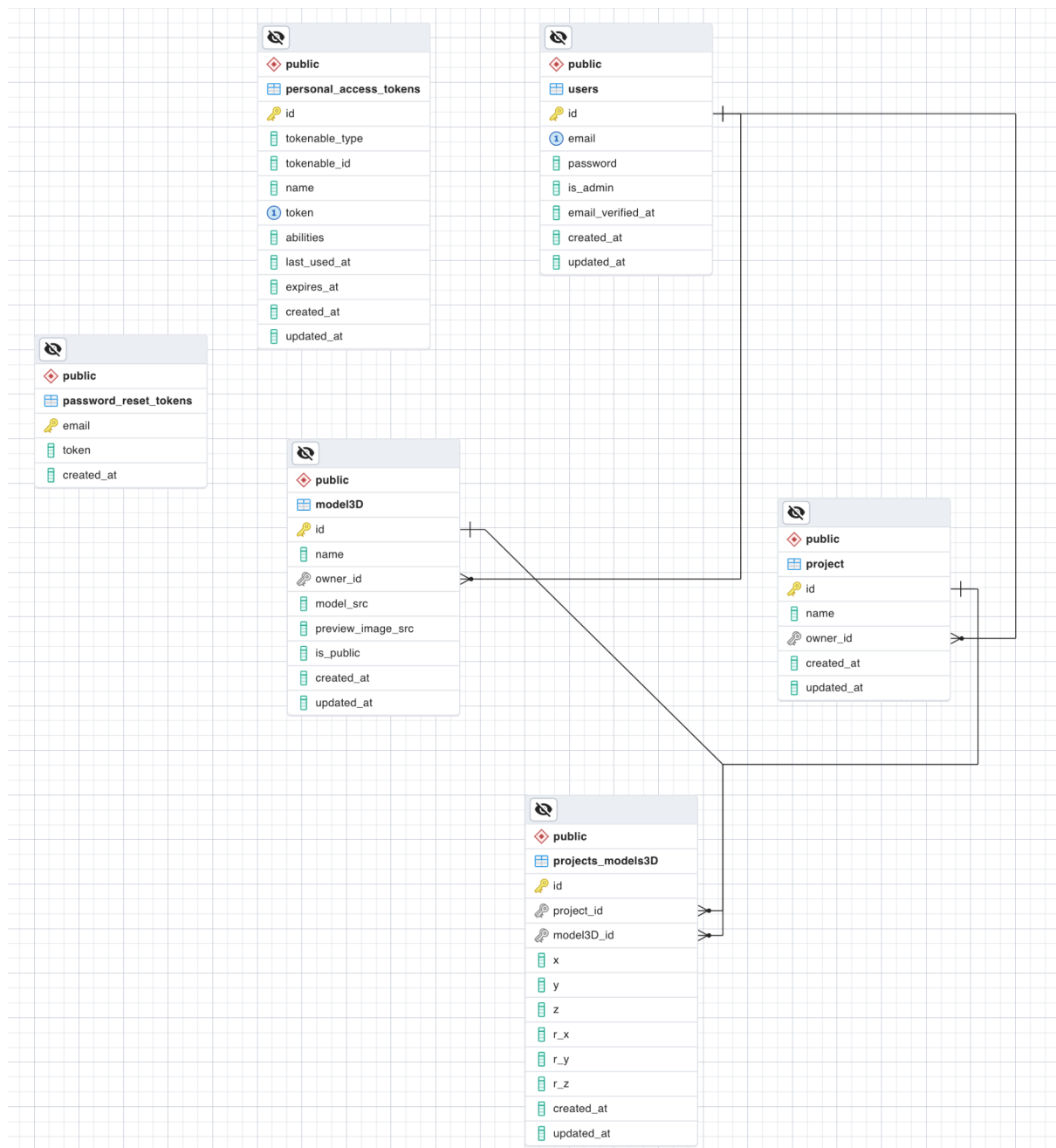
- **GitHub Webhooks:** GitHub Webhook-от е конфигуриран за да го известува Jenkins-от кога се прави „push“ на "main" гранката на репозиториумот.
- **Jenkins:** Jenkins се користи како сервер за автоматизација на CI/CD цевководот. Тој ги оркестрира и автоматизира разните фази на процесот на доставување.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



### Фази на CI/CD цевководот:

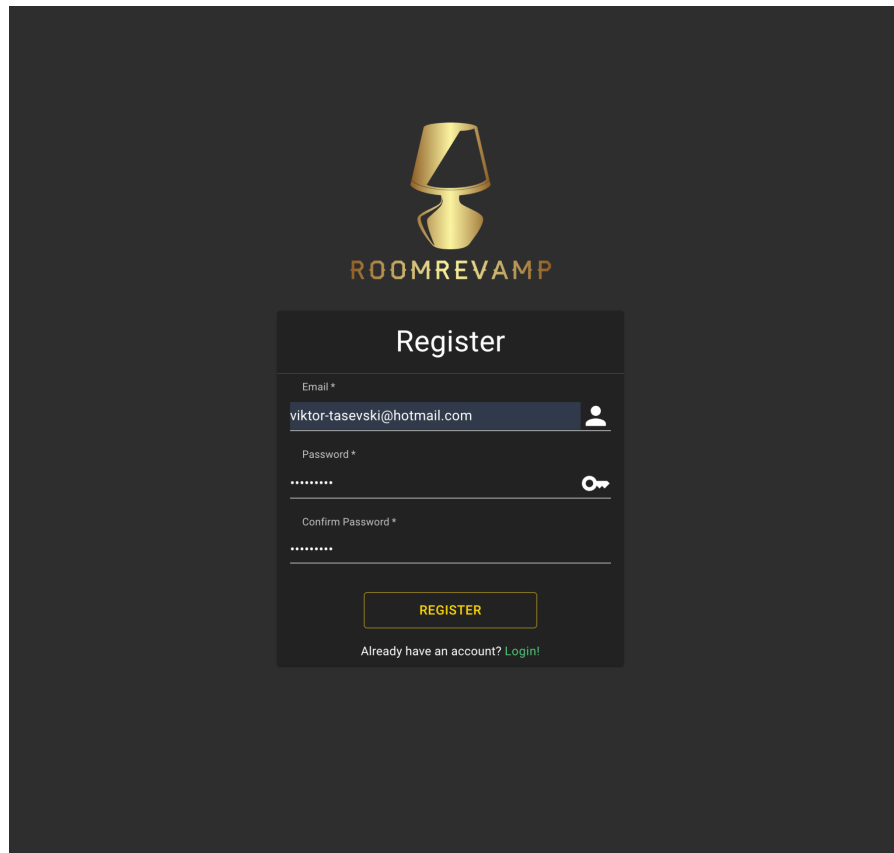
1. **Build and push backend:** CI/CD цевководот започнува со градење на бекенд кодот и создавање на Docker слика, која потоа се прикачува на DockerHub. Оваа фаза обезбедува дека најновиот бекенд код е контејнеризиран за „deployment“.
2. **Build and push frontend:** Се изградува кодот за фронтенд, и соодветна 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

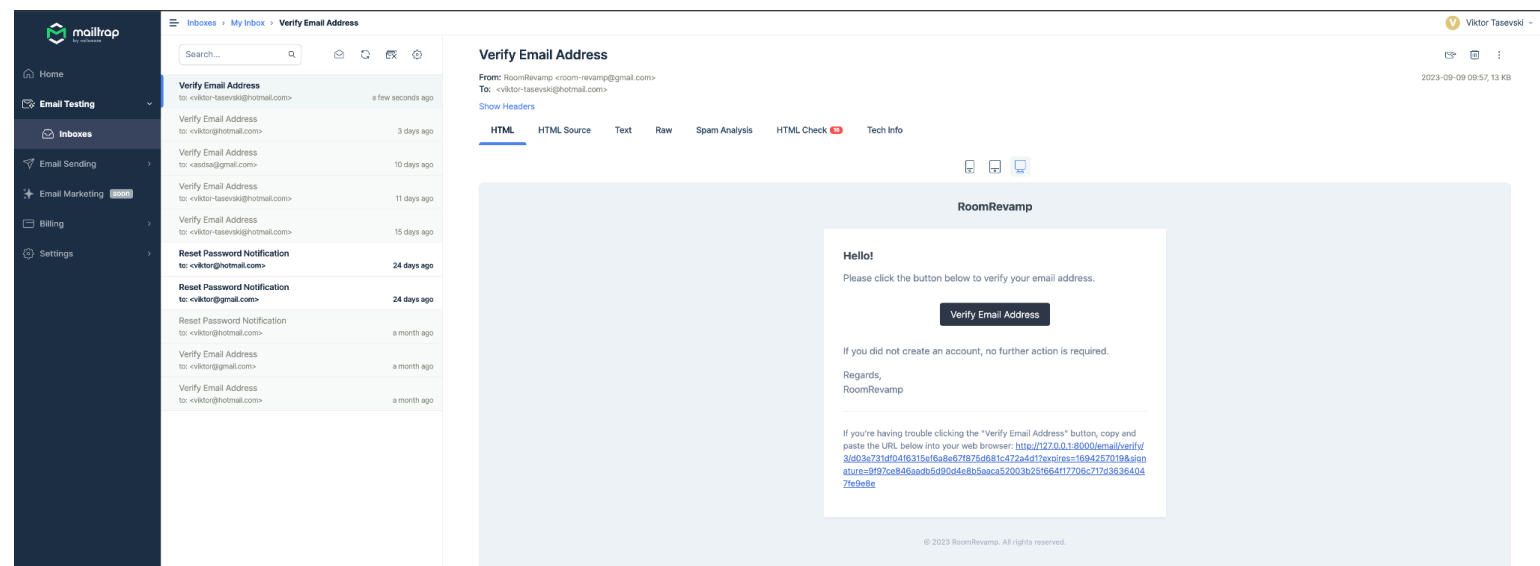
## Регистрација на корисник



*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. Доколку 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 задоволува поставените барања, не се испраќа барање до бекендот.

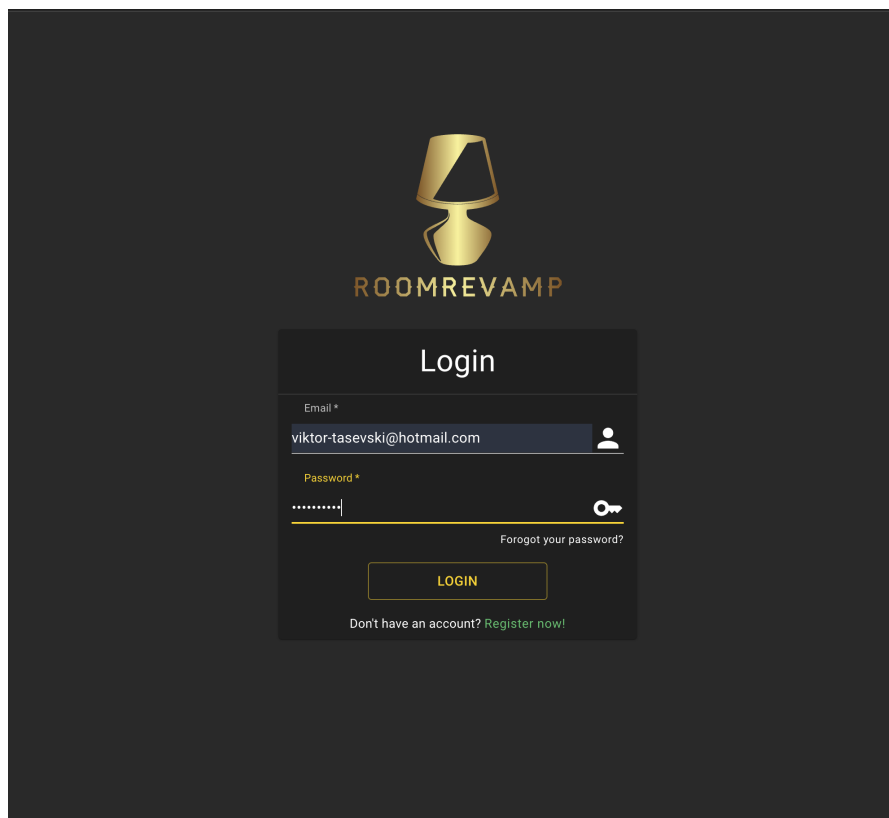
По притискање на копчето „Register“ доколку формата е валидна и барањето до бекендот е успешно, се испраќа е-маил до е-маилот на корисникот за да го потврди истиот и е пренасочен кон страната за најавување.



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 најава 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.

```

JS useCheckIfLoggedIn.js X
src > hooks > JS useCheckIfLoggedIn.js > ...
1  import { useEffect } from 'react';
2  import { useUser } from '../context/user';
3  import { API } from '../api';
4  import { setAuthHeader } from '../api/axios-instance';
5
6  const useCheckIfLoggedIn = () => {
7    const { setUser, setIsUserLoading } = useUser();
8
9    useEffect(() => {
10      async function check() {
11        const jwt = localStorage.getItem('jwt');
12        if (!jwt) {
13          setIsUserLoading(false);
14          return;
15        }
16        setAuthHeader(jwt);
17        try {
18          const res = await API.getMe();
19          setUser(res.data);
20          setIsUserLoading(false);
21        } catch (error) {
22          setAuthHeader();
23          localStorage.removeItem('jwt');
24          setIsUserLoading(false);
25          return;
26        }
27      }
28      check();
29    }, [setUser, setIsUserLoading]);
30  };
31
32  export default useCheckIfLoggedIn;

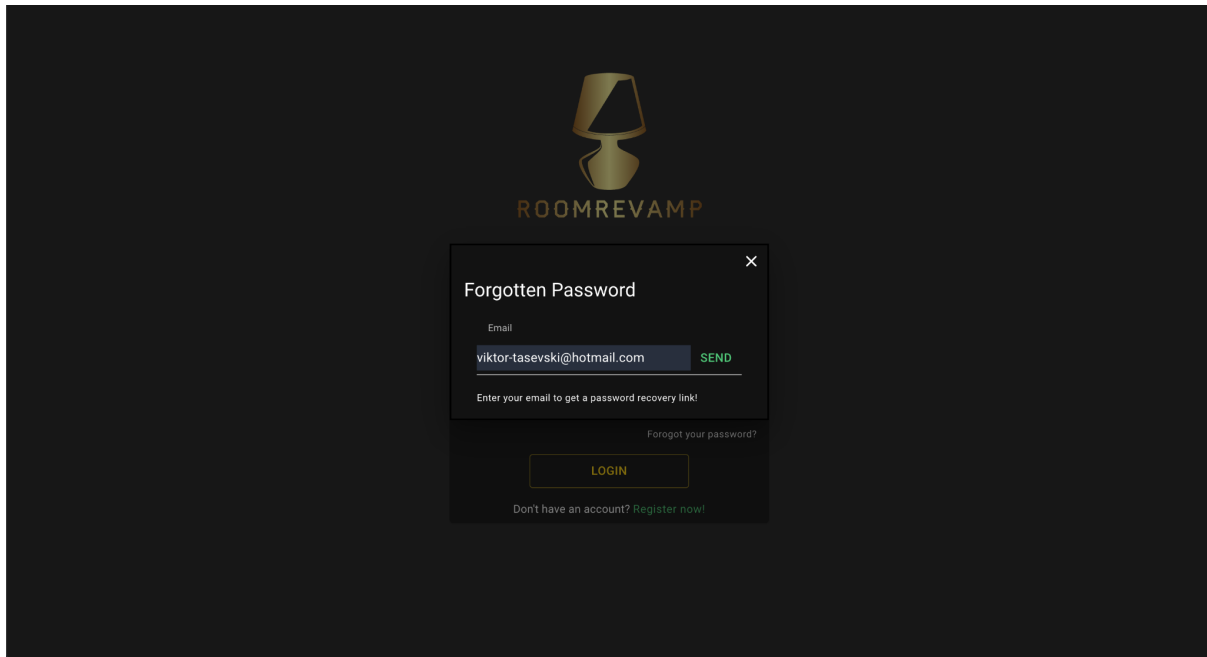
```

*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 земаме 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 axios 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 моменталниот корисник, доколку е неуспешно header-от на axios инстанцата го сетираме на празна низа од карактери и корисникот ќе биде

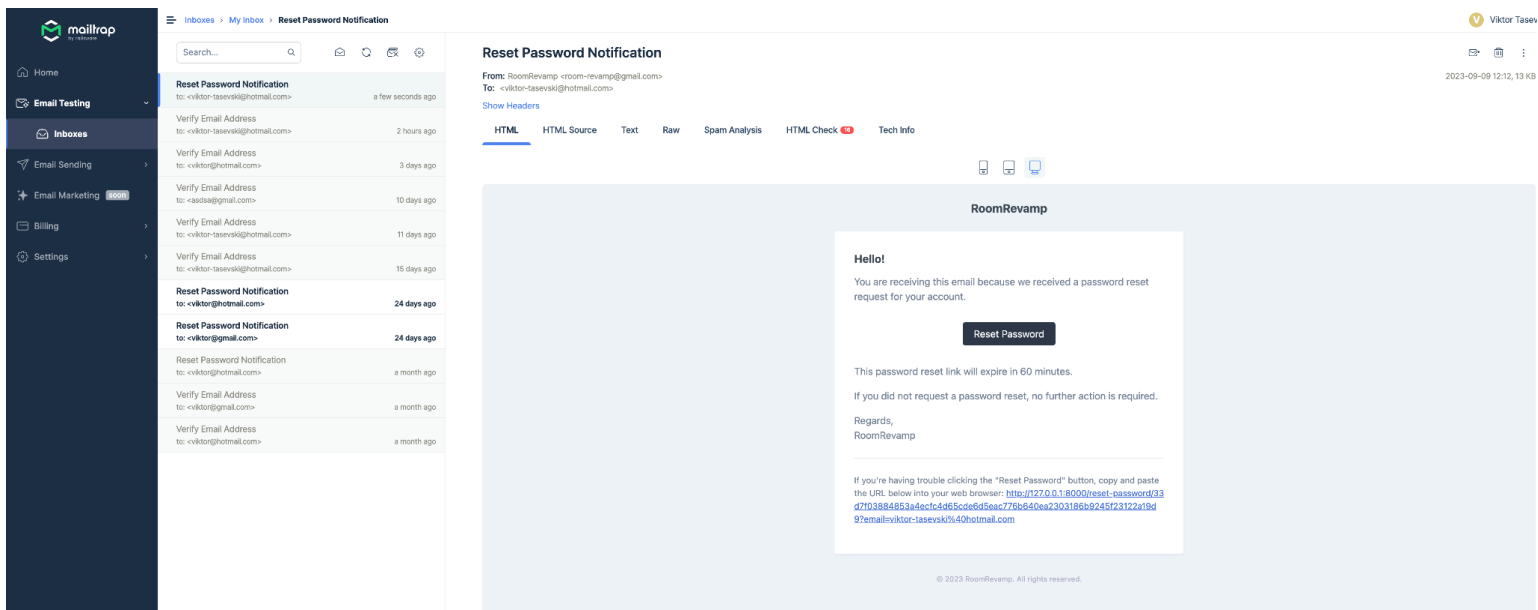
пренасочен на страната за најавување, доколку барањето е успешно, податоците за корисникот од бекендот се запамтување и корисникот е пренасочен на почетната страна.

## Заборавена лозинка



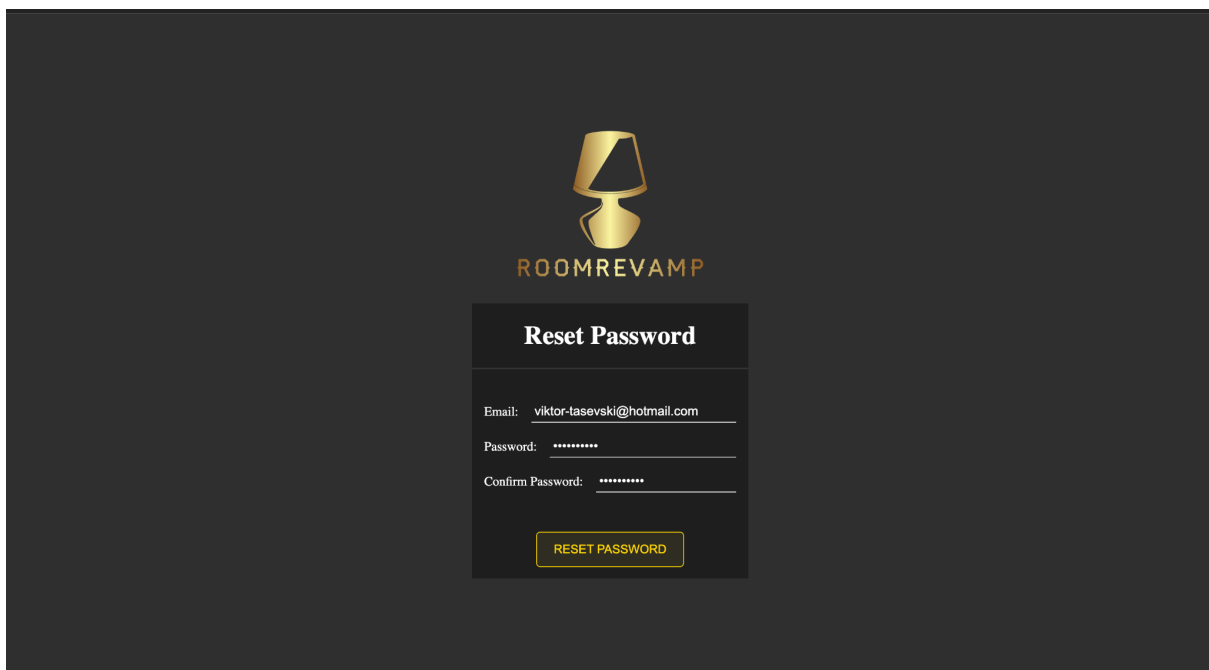
*Слика-6. Форма за барање за заборавен лозинка*

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

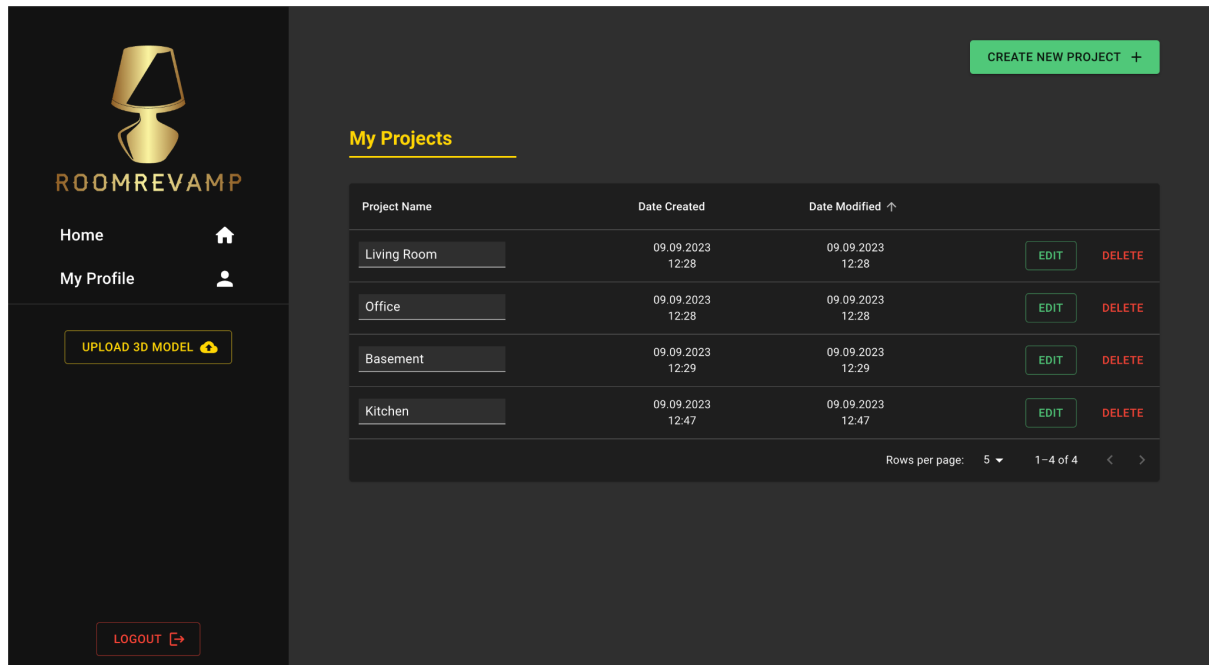
На Слика-7 можеме да го видиме е-маилот со линкот за ресетирање на лозинката. Со притискање на копчето „Reset Password“ или линкот најдоле пренасочени сме кон бекенд страната за ресетирање на лозинката.



Слика-8. Форма за ресетирање на лозинката



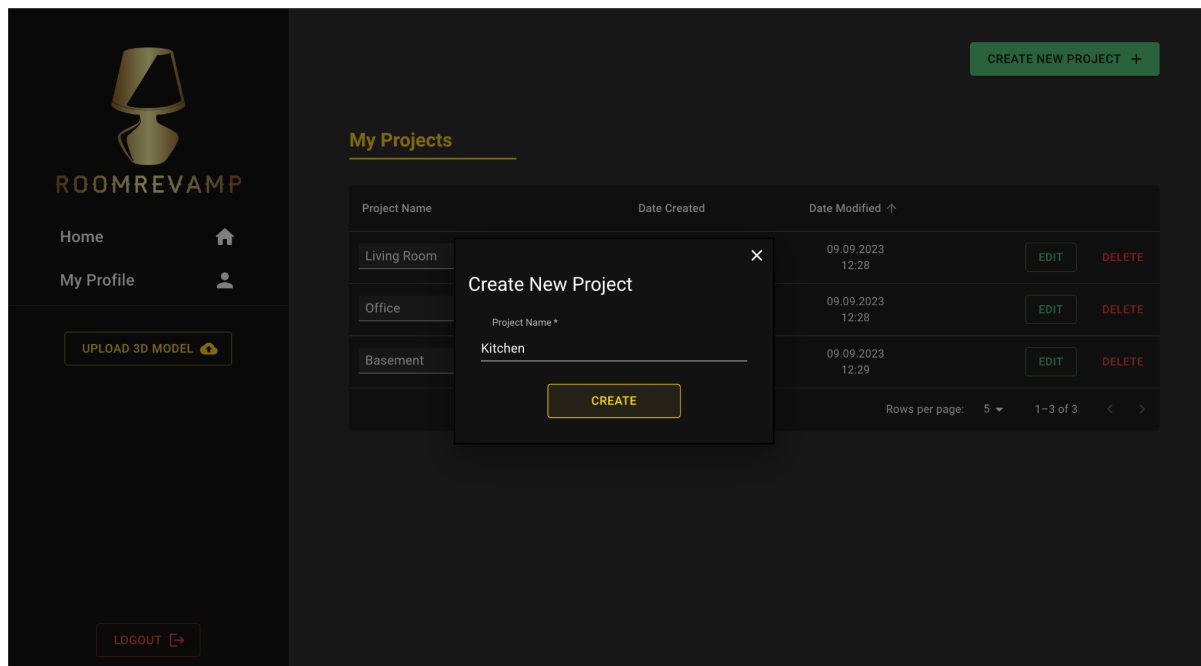
На Слика-8 потребно е да ги пополниме сите полиња на формата за ресетирање на лозинката. По притискање на копчето „RESET PASSWORD“, пренасочени сме кон фронтендот на страната за најавување.



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

Project Name	Date Created	Date Modified ↑		
Living Room	09.09.2023 12:28	09.09.2023 12:28	EDIT	DELETE
Office	09.09.2023 12:28	09.09.2023 12:28	EDIT	DELETE
Basement	09.09.2023 12:29	09.09.2023 12:29	EDIT	DELETE
Kitchen	09.09.2023 12:52	09.09.2023 12:52	EDIT	DELETE
Rows per page: 5 ▾ 1-4 of 4 < >				

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.

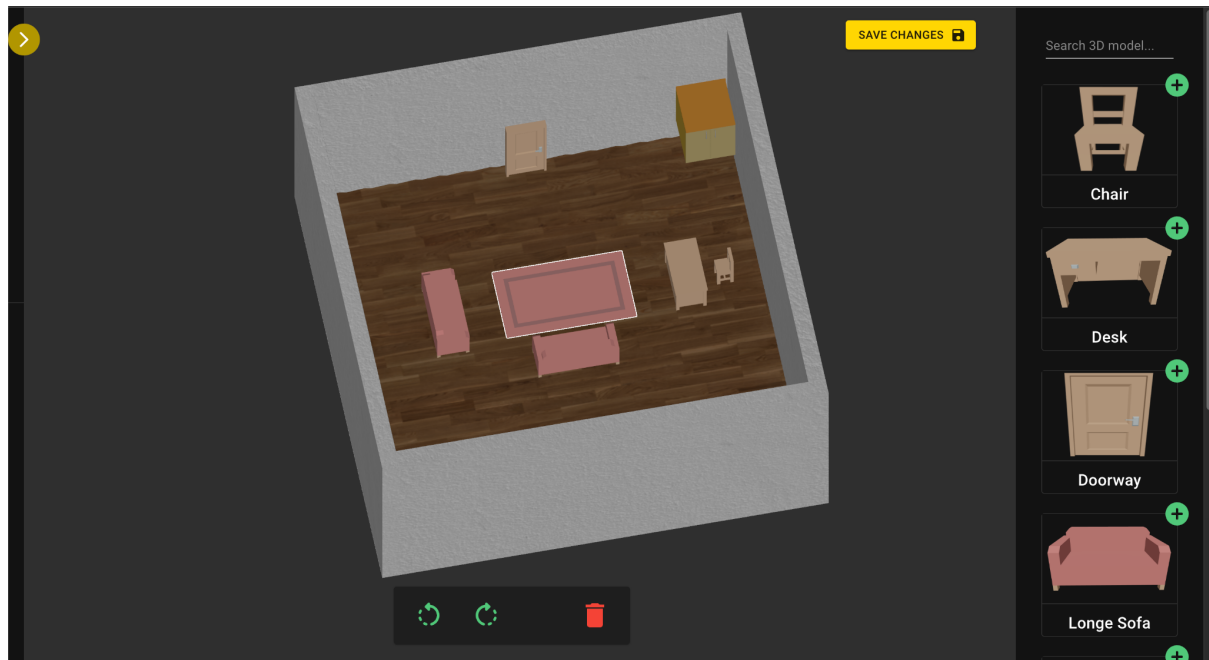
## Ажурирање и бришење на проект

Kitch	09.09.2023 12:52	09.09.2023 12:52	EDIT	DELETE
-------	---------------------	---------------------	------	--------

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 копчето „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.

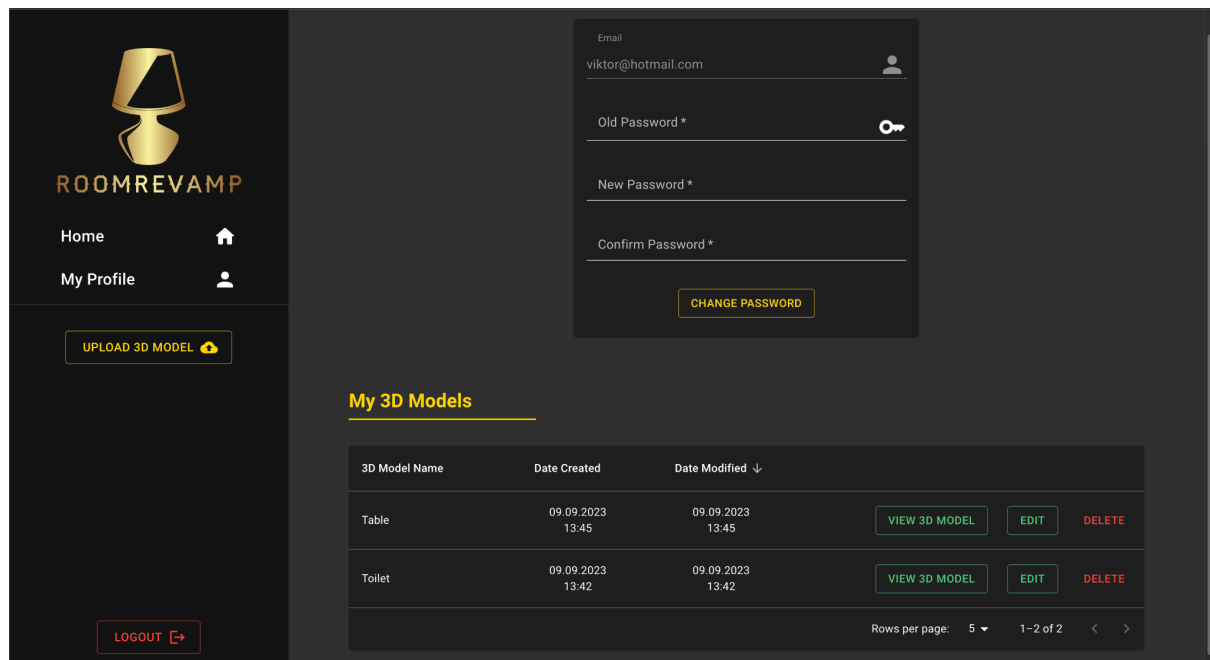
## Уредување на ентериерот



Слика-13. Страна за уредување на ентериерот

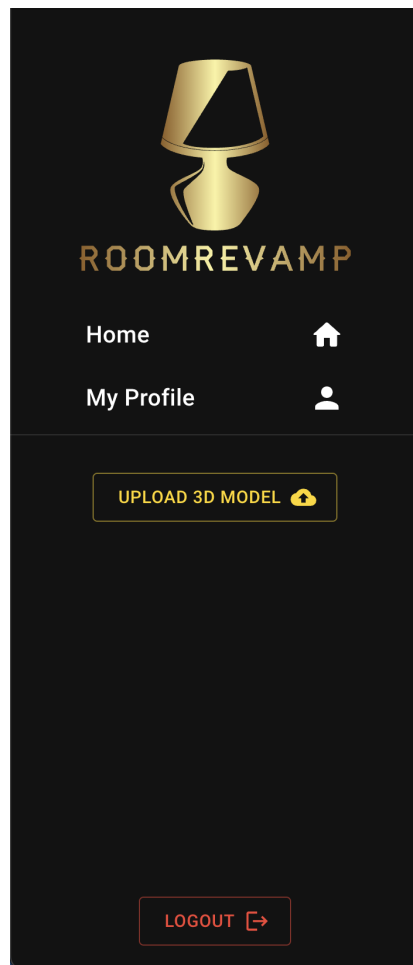
На Слика-13 е страницата за 3D уредување. Во левиот горен агол се наоѓа копче со стрелка кое го оствара страничното мени прикажано на Слика-15. Десно на екранот се наоѓа листа од 3D модели кои можеме да ги вметнеме во ентериерот со притискање на соодветното копче зелено копче со плус на некоја од картичките 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 глумчето 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“

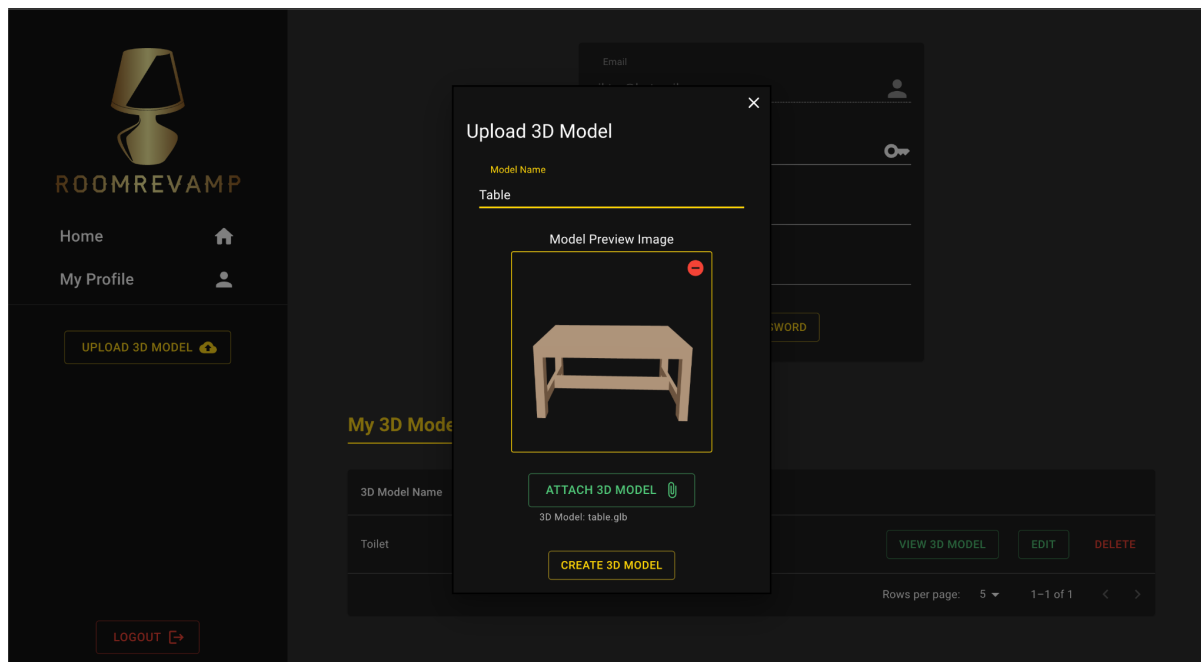
На Слика-14 имаме преглед на „My Profile“ страната. Во наредните страни ќе бидат објаснети сите функционалности.



*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 checkbox 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

3D Model Name	Date Created	Date Modified ↓			
Table	09.09.2023 13:45	09.09.2023 13:45	<a href="#">VIEW 3D MODEL</a>	<a href="#">EDIT</a>	<a href="#">DELETE</a>
Toilet	09.09.2023 13:42	09.09.2023 13:42	<a href="#">VIEW 3D MODEL</a>	<a href="#">EDIT</a>	<a href="#">DELETE</a>
Rows per page: 5 ▾ 1–2 of 2 < >					

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

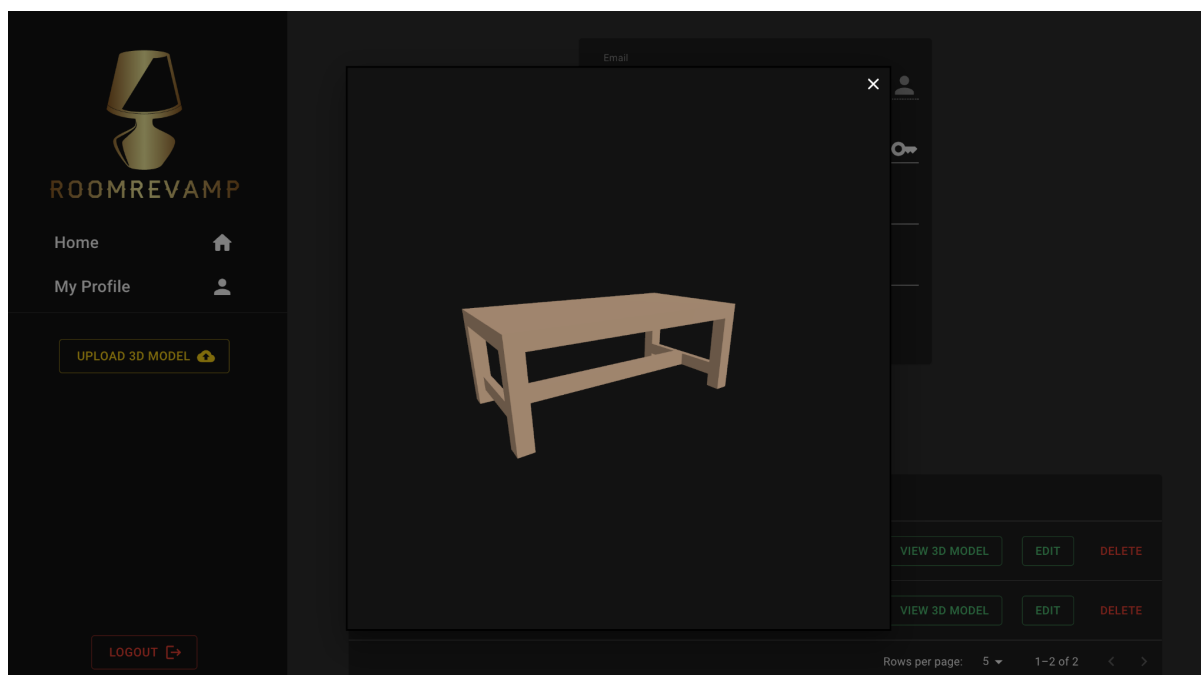
На Слика-17 имаме преглед врз нашите поставени 3D модели, опишани со името, датумот на создавање и датумот на последно ажурирање. Исто така и копчиња за 3D преглед на моделот, ажурирање и бришење.

3D Model Name	Date Created	Date Modified ↓	Is Public			
Simple Cabinet	14.08.2023 11:53	09.09.2023 13:39	Yes	<a href="#">VIEW 3D MODEL</a>	<a href="#">EDIT</a>	<a href="#">DELETE</a>
Wall Window	14.08.2023 13:02	14.08.2023 13:10	No	<a href="#">VIEW 3D MODEL</a>	<a href="#">EDIT</a>	<a href="#">DELETE</a>
Rows per page: 5 ▾ 1–2 of 2 < >						

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.

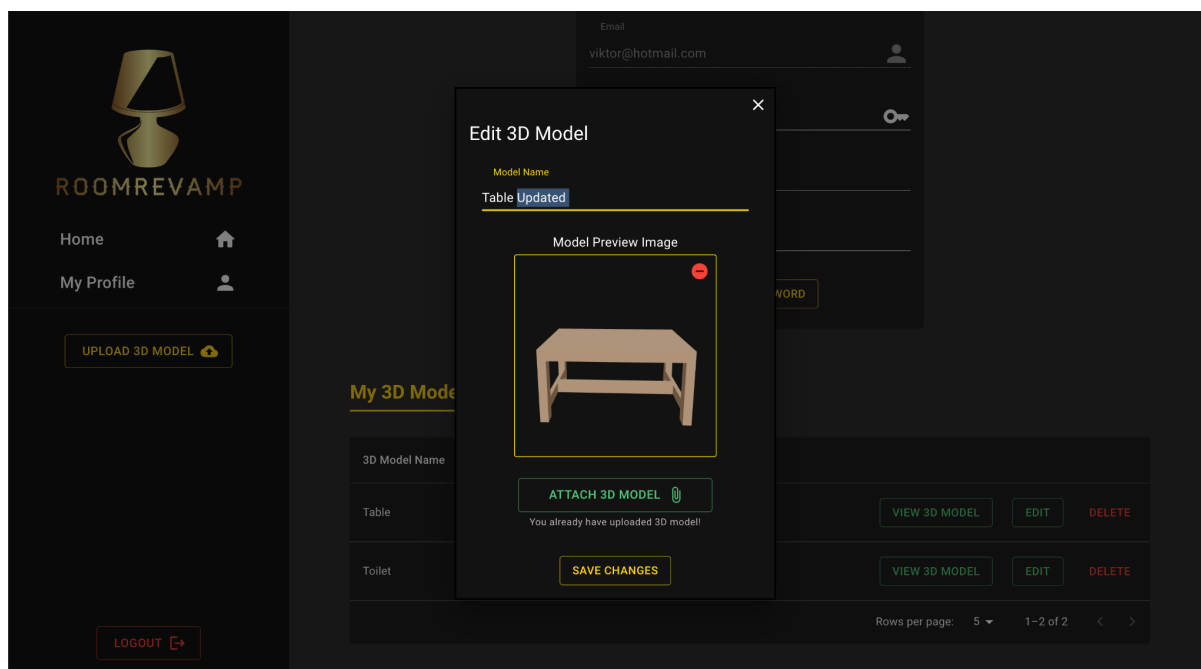
## Ажурирање и бришење на 3D модел

Table	09.09.2023 13:45	09.09.2023 13:45	<a href="#">VIEW 3D MODEL</a>	<a href="#">EDIT</a>	<a href="#">DELETE</a>
-------	---------------------	---------------------	-------------------------------	----------------------	------------------------

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 остронува од табелата на свои поставени 3D модели.



Слика-21. Ажурирање на 3D модел

## Промена на лозинка

Email

viktor@hotmail.com

Old Password \*

.....

New Password \*

.....

Confirm Password \*

.....

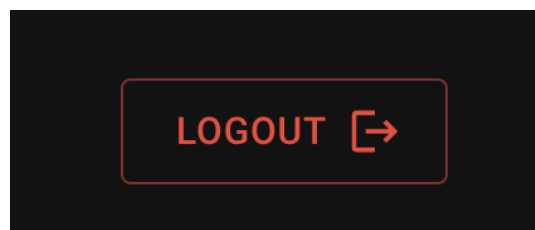
CHANGE PASSWORD

Слика-22. Форма за промена на лозинка

Во средината на екранот на страната „My Profile“ ја имаме формата за промена 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.

## Одјавување



*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.

## Проблем со ThreeJs при rerender на апликацијата

### Обрзложение на проблемот

ReactJs е изграден околу идејата на имутабилност. Кога состојбата се менува во компонента на ReactJs, тоа често резултира во поновно rerender-ирање, каде што виртуелниот DOM (VDOM) создава нова репрезентација на

корисничкиот интерфејс (UI). Потоа, овој VDOM се споредува со тековниот реален DOM, и се применуваат само потребните промени. Ова обезбедува конзистентност и ја прави состојбата во ReactJs предвидлива.

На другата страна, ThreeJs е мутабилен. Кога модификуваме објект во ThreeJs (како неговата позиција, ротација или некое друго својство), ние директно ја менуваме неговата состојбата. Не се создава целото сценарио или објект секој пат кога настанува некоја промена.

Кога во ReactJs компонентите се rerender-ираат, тие може повторно да ги создадат своите деца, своите својства и другите поврзани објекти. Ако не внимаваме како е поставена ThreeJs сцената во ReactJs компонента, ненамерно може ќе создаваме нови ThreeJs објекти при секој render.

Ова може да доведе до неочекувано однесување (како неочекувано движење на 3D елементите - како во мојата апликација), проблеми со перформанси (зависни од константното создавање на објекти и собирањето на отпадоци) и преминување на лимитот на меморија (ако референците до старите ThreeJs објекти не се правилно ослободени).

ReactJs примарно го ажурира DOM-от, додека ThreeJs го ажурира WebGPU. Кога имаме компоненти на ReactJs што ги контролираат или комуницираат со сцените на ThreeJs, треба да бидеме внимателни за овие два посебни механизми за ажурирање и да се погрижиме дека не се во конфликт.

ReactJs има свој систем за настани, кој може да се меша во справувањето со настаните на ThreeJs. На пример, ако користиме ReactJs за „drag-and-drop“ во сцена на ThreeJs, ажурирањата на состојбата и rerender-ирањето од ReactJs може да предизвика неправилно пресметување на позицијата во Three.js, што доведува до непредвидливо однесување.

## **Совети за надминување на проблемот**

**Користење на референци:** Со користење на React refs, можеме да создадеме постојана референца до нашите ThreeJs објекти кои нема да се менуваат меѓу

renders. Ова може да помогне да се осигураме дека истите ThreeJs објекти се користат низ render-от.

**Мемоизација и оптимизации:** Со користење на вградените механизми за оптимизација на ReactJs, како React.memo, useMemo и useCallback, за да се спречат непотребни renders или креирања на објекти.

## 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](#), [windows](#), [linux](#) ), 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](#), [windows](#), [linux](#) ), бидејќи содржи голем дел од потребните алатки.

Потребно е да имате инсталирано k3d ( <https://k3d.io/v5.6.0/#installation> ), kubectl ( [macOs](#), [windows](#), [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 и бекенд 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 exec -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**“ искачаеме 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/>.

```
[viktortasevski@MacBook-Pro-4 ~ % cat /etc/hosts
##
# Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting. Do not change this entry.
##
127.0.0.1        localhost room-revamp.com api.room-revamp.com
255.255.255.255 broadcasthost
::1             localhost
# Added by Docker Desktop
# To allow the same kube context to work on the host and the container:
127.0.0.1 kubernetes.docker.internal
# End of section
```