

УНИВЕРЗИТЕТ У БЕОГРАДУ  
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ



# **АПЛИКАЦИЈА ЗА УПРАВЉАЊЕ РАДНИМ ПРОСТОРОМ**

## Мастер рад

Ментор:  
проф. др Милош Цветановић

Кандидат:  
Марина Вучковић 2020/3261

Београд, Септембар 2022.

# САДРЖАЈ

<b>САДРЖАЈ .....</b>	<b>I</b>
<b>1. УВОД.....</b>	<b>1</b>
<b>2. АНАЛИЗА ПРОБЛЕМА .....</b>	<b>3</b>
2.1. НЕОПХОДНЕ ФУНКЦИОНАЛНОСТИ .....	3
2.2. ПОСТОЈЕЋИ СИСТЕМИ .....	4
2.2.1. <i>Envoy Desks</i> .....	4
2.2.2. <i>Condeco</i> .....	4
2.2.3. <i>OfficeSpace Software App</i> .....	4
2.2.4. <i>Spacewell Workplace</i> .....	5
2.2.5. <i>HotDesk+</i> .....	5
<b>3. ЗАХТЕВИ ЗА РЕАЛИЗАЦИЈУ СИСТЕМА.....</b>	<b>6</b>
3.1. КОРИСНИЧКИ ЗАХТЕВИ.....	6
3.1.1. <i>Заједничке функционалности</i> .....	6
3.1.2. <i>Функционалности доступне администраторима</i> .....	6
3.1.3. <i>Функционалности доступне запосленима</i> .....	9
3.2. ОПИС КОРИШЋЕНИХ ТЕХНОЛОГИЈА .....	10
3.2.1. <i>Клијентска страна</i> .....	10
3.2.2. <i>Серверска страна</i> .....	11
3.2.3. <i>База података</i> .....	12
3.2.4. <i>Платформе за извршавање апликације</i> .....	12
<b>4. ОПИС РАДА СИСТЕМА.....</b>	<b>14</b>
4.1. КОРИСНИЧКИ ЕКРАНИ АДМИНИСТРАТОРА .....	14
4.1.1. <i>Главни кориснички екран администратора</i> .....	14
4.1.2. <i>Канцеларије</i> .....	15
4.1.3. <i>Пројекти</i> .....	16
4.1.4. <i>Запослени</i> .....	16
4.1.5. <i>Распоред седења</i> .....	18
4.1.6. <i>Статистика попуњености</i> .....	20
4.2. КОРИСНИЧКИ ЕКРАНИ ЗАПОСЛЕНОГ .....	21
4.2.1. <i>Колеге и њихова организација по пројектима</i> .....	21
4.2.2. <i>Резервација радног места</i> .....	21
4.2.3. <i>Приказ распоиреда седења по канцеларијама</i> .....	24
<b>5. РЕАЛИЗАЦИЈА СИСТЕМА.....</b>	<b>26</b>
5.1. РЕАЛИЗАЦИЈА СЕРВЕРСКЕ СТРАНЕ .....	26
5.1.1. <i>Модел података</i> .....	26
5.1.2. <i>Дефинисање рута</i> .....	27
5.1.3. <i>Конфигурација сервера</i> .....	28
5.1.4. <i>Алгоритам расподеле</i> .....	28
5.2. РЕАЛИЗАЦИЈА КЛИЈЕНТСКЕ СТРАНЕ .....	32
5.2.1. <i>Компоненте</i> .....	32
5.2.2. <i>Комуникација клијентске и серверске стране</i> .....	33
<b>6. ЗАКЉУЧАК.....</b>	<b>35</b>
<b>ЛИТЕРАТУРА.....</b>	<b>37</b>

<b>СПИСАК СКРАЋЕНИЦА .....</b>	<b>38</b>
<b>СПИСАК СЛИКА.....</b>	<b>39</b>

# 1. Увод

Пандемија изазвана вирусом Ковид-19 имала је утицаја на све сфере живота, па тако и на начин пословања и на навике запослених. Током две године пандемије, рад од куће је постао нова свакодневица. Са променом епидемиолошке ситуације и укидањем рестриктивних мера уведених да би се спречило ширење пандемије, многе фирме су пожелеле да се врате на стари модел рада. Међутим, испоставило се да већина запослених не жели свакодневно да буде у канцеларији. С друге стране, део запослених сматра да се искључивим радом од куће губи социјални фактор, тимским дух и комуникационе вештине. Као резултат тога, компаније су прихватиле чињеницу да морају да примене другачији принцип рада, те се хибридни модел наметнуо као нови стандард.

Хибридни модел рада је организација пословања која представља комбинацију рада у канцеларији и рада на даљину. Запослени имају могућност да бирају којим данима ће долазити у канцеларију, а када ће радити од куће. Овај модел отвара могућности великих уштеда у простору и ресурсима, јер се неће дешавати да сви запослени буду физички присутни истовремено. С друге стране, јавља се и изазов како успешно управљати расположивим ресурсима.

У овој тези приказана је апликација која аутоматизује управљање радним простором код хибридног модела рада. Апликација је реализована применом савремених технологија у области интернет пословања. Циљ тезе је да се применом имплементираних решења омогући запосленима да имају резервисано место за рад у оквиру компаније за оне дане у радној недељи када желе да посао обављају из канцеларије. Такође, имплементиран је и алгоритам расподеле запослених који групише све људи из истог тима да буду заједно у канцеларији. С друге стране, администраторима је омогућена одговарајућа статистика попуњености компаније. На основу ње може да се донесе закључак да ли су у дефициту или суфициту са ресурсима, и у складу с тим предузму одговарајуће кораке. Очекивани допринос тезе је имплементација кода који може једноставно да се одржава и унапређује у будућности и који даје одговоре на изазове управљања радним простором.

Теза је организована у више поглавља. Поглавље два описује шта се све подразумева да буде подржано у једном систему овог типа да би задовољио потребе корисника. Дат је и преглед постојећих апликација на тржишту које су наишле на највећи одзив клијената. Такође су истакнуте и њихове предности и мане.

Поглавље три даје преглед главних корисничких захтева који су подржани у оквиру апликације. Такође је дат осврт на коришћене технологије које су омогућиле реализацију система. Представљене су и библиотеке, платформе и алати који су олакшали имплементацију одређених корисничких захтева.

Поглавље четири описује рад система и његове главне функционалности. Приказане су и слике корисничких екрана, чиме се на најефикаснији начин стиче увид у рад целокупног система.

У поглављу пет дата је реализација система и имплементациони део. Приложени су и објашњени круцијални делови кода, као и они коју су били најизазовнији приликом израде саме апликације. Представљени су делови базе података, идејно решење алгоритма расподеле као и начин комуникације између делова система.

## 2. АНАЛИЗА ПРОБЛЕМА

Хибридни модел рада постојао и пре пандемије. Међутим, незнатан број компанија је видео његове предности, те је био слабо примењиван. На самом почетку пандемије све делатности су наишле на велики изазов како да одрже своје пословање, а да не угрозе здравље запослених. Стога су послови који могу да се обављају и са удаљених локација у потпуности пренети да се обављају од куће. С повременим побољшањима епидемиолошке ситуације модел рад је постајао све више комбинован. Наиме, увек постоји минимум активности које је неопходно урадити уживо тј. на терену, када ситуација то дозвољава.

Током првог периода удаљеног модела рада чинило се да су запослени задовољнији. Поред посла имали су времена да се посвете својим породицама, и нису губили време на транспорт до истог. Штавише, показало се и да је целокупна ефикасност извршавања задатака већа. Међутим, како ситуација није јењавала, а рад од куће се продужавао, на површину су испливале и негативне стране оваког модела. На дуже стазе се показало да део запослених нема адекватан простор и за живот и за рад. Као додатни изазов појавио се недостатак комуникације са колегама и члановима тима.

Пандемија је у многоне променила свет, па тако ни начин пословања није остао имун. Поред традиционалног рада из канцеларија, у пракси је виђен и удаљени рад. То је имало утицаја и на промену свести и приоритета запослених. Њихово здравље и добробит им је постало важније него раније. Такође, постали су склонији мењању послова.

Испоставило се да већини запослених не одговара у потпуности ни рад из канцеларије ни удаљени модел рада, већ да је оптимална флексибилност нађена у хибридном раду. Схвативши да су се очекивања запослених потпуно променила, компаније су морале да се прилагоде да би задржале тренутни кадар, и успешно запошљавале нови. Све то је подстакло и развој софтвера који би оптимизовао капацитете компаније. Наиме, нестала је потреба да сваки запослени има своје место. Управо ово поглавље детаљније описује које су све то функционалности које мора да подржи једна апликација оваког типа. Такође, представљен је однос понуда постојећих апликација и клијентске потражње.

### 2.1. Неопходне функционалности

Да би једна апликација за управљање радним простором успела у данашње време и била конкурентна на тржишту, неопходно је да пре свега нуди квалитет, сигурност и перформансе високог нивоа. Такође је потребно идентификовати оптималан скуп функционалности тако да спектар могућности за запослене и администраторе буде широк, а опет ефикасан, користан и интуитиван. Сходно томе, биће описане које су то могућности које мора да има апликација овог типа да би задовољила потребе корисника.

Основне функционалности оваког система су резервисање радних места и алгоритам расподеле запослених. Пре свега је неопходно дефинисати на који начин ће запослени резервисати своја радна места и на који начин ће бити распоређени. Постоји више начина приступања овом проблему. Нека од њих су да запослени може да резервише свако место, те се у овом приступу запослени заправо сами распоређују. Даље је могуће да алгоритам у

оквиру апликације дозвољава запосленим да бирају неко од препоручених места. Још једна од могућности је да се запослени изјашњавају путем анкете када желе да раде из канцеларије, а потом да их алгоритам распоређује по неком критеријуму. Управо је поменути приступ примењен у овом раду.

Што се алгоритма расподеле запослених тиче, узета је претпоставка да запослени који су на истом пројекту највише комуницирају. Стога алгоритам распоређује запослене тако да људи који су на истом пројекту буду у истој канцеларији.

Следећа функционалност која је потребна јесте да распоред седења буде видљив како администраторској страни тако и запосленима. Такође се показало као добра пракса да постоји опција претраге распореда долазака изабраног колеге као и његовог места рада за одређени датум. Тиме се постиже да се лако проналази тражени колега када је потребна комуникација са њим. Да би администратори могли да прилагоде своје капацитете потражњи, од значаја је да буде имплементиран и преглед попуњености фирме и канцеларија.

## **2.2. Постојећи системи**

Надаље ће бити приказане неке од апликација која су наишле на највећи одзив корисника. Биће прокоментарисане неке од функционалности због којих су управо ове апликације задовољиле потребе компанија и њихових запослених, и тиме се издвојиле на тржишту.

### **2.2.1. *Envoy Desks***

Ово је једна од најпопуларнијих апликација за управљање радним простором компанија. Издваја је добро осмишљен проширени сет функционалности. Апликација нуди запосленима могућност да организују састанке позивајући колеге и клијенте да дођу до њиховог радног места у договорено време. Пријава на радно место омогућена је путем мобилног уређаја. Приликом доласка на посао свако појединачно попуњава анкета каквог је здравственог стања, чиме су предузете све неопходне мере безбедности. Апликација има имплементиран и систем навигације којим упућује запосленог од његове тренутне локације до места састанка. Такође, има и развијен систем пријема пошиљке, у којем запослени буду обавештени када њихова наруџбина стигне на адресу компаније.

### **2.2.2. *Condeco***

Поред резервације места за рад, ова апликација пружа и могућност резервације паркинга, ормарића за одлагање, просторија у којима нема буке ни ометања, као и просторија за рекреацију. За разлику од претходне апликације, пријава на радно место није омогућена преко мобилног уређаја, већ преко персонализованих картица. Свакако и ово решење омогућава да се у реалном времену испрати попуњеност компаније. Уколико се запослени не пријави на резервисано место у претходно одређеном временском оквиру, место се ослобађа, а самим тим расте и искоришћеност ресурса.

### **2.2.3. *OfficeSpace Software App***

Поред интуитивног и ефикасног резервисања места за рад и дељења локације са колегама, ова апликација нуди доста олакшања руководиоцима објекта. Наиме, чак и када нису физички присутни могу да управљају ресурсима. Доступни су им подаци у реалном

времену представљени кроз тлоцрт и извршне командне табле. Такође, софтвер пружа и сценарије који омогућавају прављење будућег распореда седења у тимовима.

#### **2.2.4. *Spacewell Workplace***

Од напредних функционалности апликација нуди тлоцрт радног простора у реалном времену. Тиме је омогућено да се виде радни услови сваке просторије фирме по критеријуму попуњености и буке, као и температуре просторије, влаге, сунчане стране. Да би услови рада били оптимални, апликација прати тренутне препоручене мере понашања ради сузбијања заразе, те контролише и да ли су их сви чланови фирме испоштовали. Додатно, подржана је и опција наручивања хране и пића, и праћења наруџбине.

#### **2.2.5. *HotDesk+***

Апликација интегрише све састанке и резервације кроз *Google G Suit*, *Microsoft 365* и *Teams* календаре. Попут претходних апликација пријава на радно место је реализована путем мобилног уређаја, а такође постоји и могућност резервације паркинг места. Простор може да ради по принципу да свако има своје место, или да места буду додељена онда када их запослени затраже.



## 3. ЗАХТЕВИ ЗА РЕАЛИЗАЦИЈУ СИСТЕМА

У оквиру овог поглавља биће представљени главни кориснички захтеви који су имплементирани у оквиру самог система. Потом ће бити описане и технологије које су употребљене у реализацији апликације. Биће истакнуте и библиотеке, платформе и алати које су омогућиле једноставну имплементацију одређених функционалности. Приказани су и концепти, архитектура и протоколи коришћени за изградњу система.

### 3.1. Кориснички захтеви

У наредним секцијама биће описани реализовани кориснички захтеви. Апликација је намењена за два типа регистрованих корисника: администраторе и запослене. Сходно томе разликујемо заједничке функционалности, функционалности доступне администраторима и функционалности доступне запосленима. Надаље ће сваки од наведених скупова корисничких захтева бити детаљно објашњен.

#### 3.1.1. *Заједничке функционалности*

Приликом покретања апликације доступне су функционалности које су заједничке за оба типа корисника. Они имају могућности пријаве у систем тј. логовања ако су већ регистровани, одн. могућност регистрације уколико први пут приступају апликацији. Након успешне аутентикације сви корисници имају опцију да се у сваком тренутку одјаве из система и напусте рад. Оно што је такође заједничко за оба типа корисника у сваком тренутку је да имају могућност да се свака страница прикаже на оном језику који је тренутно изабран од понуђених.

Регистрација је реализована тако да корисници у одговарајућа поља уносе име и презиме, имејл адресу, лозинку и потврду лозинке. Такође постоји и поље за потврду да ли је у питању администратор, те корисници у складу са својим типом треба да обележе поље. У случају да корисник не унесе података, или унесе невалидне, исписаће се одговарајућа порука.

Што се логовања у систем тиче, оно је могуће за све регистроване кориснике. Они се логују у апликацију тако што уносе своју имејл адресу и лозинку у одговарајућа поља. У случају уноса неисправне имејл адресе или лозинке исписују се адекватне поруке. У супротном се корисник преусмерава на почетну администраторску страну, одн. на почетну страну запосленог, у зависности од типа корисника.

#### 3.1.2. *Функционалности доступне администраторима*

У овом одељку ће бити приказани кориснички захтеви који су доступни администраторској страни. Они су везани за следеће ентитете у систему: компаније, канцеларије, пројекти, запослени, распоред седења и статистика попуњености компаније. Тим редоследом ће бити и детаљније описани у наставку.

#### *i) Компаније*

Администратор има могућност креирања једне или више компаније. Да би креирао компаније треба да унесе следеће информације:

- Име компаније
- Адреса
- Контакт имејл адреса
- Контакт телефон
- Радно време

Приликом успешне аутентикације администратор се одводи на почетну администраторску страну. Ту се налази списак компанија за које је задужен. Избором неке од њих постају му доступне све опције везане за ентитете дате у наставку. Такође има и опције ажурирања и брисања компаније.

#### *ii) Канцеларије*

Након селектовања жељене компаније администратору постаје доступна опција креирања канцеларије за изабрану компанију. Приликом креирања администратор треба прво да унесе име канцеларије, спрат, и број радних места која би могла да стану по дужини и ширини канцеларије. Потом се избором орјентације радног места и превлачењем одговарајуће иконице, или једноставним кликом на жељено поље шеме, попуњава канцеларија и прави се њен тлоцрт.

Када је компанија селектована видљиве су и све њене канцеларије. Канцеларије су груписане и приказане по спратовима. Избором неке од њих постају доступне опције приказа додатних информација о канцеларији, њеног тлоцрта као и опција ажурирања и брисања исте.

#### *iii) Пројекти*

Груписање запослених по пројектима има важну улогу у овом систему. Поменути алгоритам расподеле функционише по принципу да запослене групише по канцеларијама тако да запослени који су задужени за исти пројекат буду заједно у канцеларији. Стога, администраторска страна има могућности прегледа свих пројеката, као и креирања, ажурирања и брисања истих. Такође, постоје и опције додавања и уклањања запослених са пројеката.

#### *iv) Запослени*

Управа компаније има реализован и кориснички захтев прегледа запослених. Ова функционалност се састоји из два дела: преглед запослених који су регистровани и преглед запослених који су позвани у апликацију, али се нису још регистровали.

Што се тиче прегледа регистрованих запослених, он се приказује у виду листе, одн. по редовима и колонама уколико има доста запослених. За сваког запосленог се приказује његово име и презиме, као и имејл адреса. Поред тога, наглашено је и за који је пројекат задужен.

Преглед запослених који су позвани у апликацију а нису се још регистровали је такође приказан у виду листе. Листа садржи њихове имејл адресе. Ту постоји и опција позива запослених у апликацију.

Позивање запослених у апликацију је имплементирано тако да се аутоматски шаље имејл новом кориснику када администратор потврди форму. Наиме, слање имејла је реализовано у виду форме, где администратор треба да унесе име и презиме будућег запосленог, као и његову имејл адресу. У остатку форме налазе се поља као што је препоручен наслов и текст поруке. У препоручени текст се динамички уписује име и презиме будућег запосленог на почетку имејла, одн. име и презиме администратора у потпису. Препоручени наслов и текст могу и да се ручно измене уколико се укаже потреба.

Након потврде поменуте форме од стране администратора, позивни имејл се шаље будућем запосленом. Имејл који стигне има и линк ка апликацији, који одводи корисника на страну за регистрацију. Након успешне аутентикације, корисник је аутоматски додат у компанију.

#### *v) Распоред седења*

Следећа реализована функционалност за администраторску страну је приказ распореда седења запослених. Наиме, администратор из календара бира датум за који жели да погледа распоред. Да би се запосленима оставила максимална флексибилност, систем је направљен тако да су пријаве за наредни радни дан запосленима доступне до 20h текућег дана. Дакле, алгоритам сваког радног дана у 20h генерише распоред за наредни радни дан. Самим тим, ова функционалност представља историјат распореда седења. На основу ње управа компаније може да размотри који то запослени практикују да раде из канцеларије, који удаљено, и да ли капацитети неких канцеларија треба да се ажурирају.

Након избора жељеног датума из календара, администратору се учитава и распоред седења. Он има преглед канцеларија груписаних по спратовима, и селектовањем једне од њих учитава се распоред за жељену канцеларију. Ту се налази списак запослених којима је алгоритам доделио ту канцеларију за изабрани датум, гледајући да запослени са истог пројекта буду у истој канцеларији. Такође је и за сваку канцеларију приказано колико је радних места било неискоришћено.

#### *vi) Статистика попуњености компаније*

Наредна реализована функционалност за администратора је статистика попуњености компаније по данима. Наиме, када се изабере радна недеља, учитава се и анализа попуњености за њене радне дане. Она је визуелно представљена путем хистограма, где постоје две врсте колона.

Прва колона представља број радних места која су била заузета одређеног дана. Друга колона приказује колико је било запослених који су хтели да тај дан да раде из просторија компаније, али су у тренутку њихове пријаве сва места већ била попуњена. Она је и истакнута црвеном бојом, да би дала додатно визуелно упозорење администратору када капацитети компаније нису задовољили потребе запослених.

Анализом поменутог графикана управа компаније добија јасну визуелну представу какву су навике и тенденције запослених. У складу с тим, уколико део капацитета никад није попуњен, те је друга колона увек празна, требало би да се размисли да се ресурси редукују. С друге стране, ако су капацитети најчешће попуњени до максимума, и вредност друге колоне често различит од нуле, тада треба размотрити евентуално повећање број радних места и ресурса.

### **3.1.3. Функционалности доступне запосленима**

У овом одељку представљени су главни кориснички захтеви који су доступни запосленима. Функционалности које ће бити детаљније описане су: приказ колега, приказ сопственог календара и резервација радног места, преглед распореда долазака изабраног колеге и приказ распореда седења.

#### *i) Приказ колега*

Запослени пре свега има могућност прегледа свих својих колега. Ова функционалност је реализована тако да су сви запослени излистани и груписани по задужењима одн. пројектима на којима раде. Поред имена и презимена колеге, назначена је и имејл адреса. На овај начин запослени одмах има увид и у контакт колеге, те може да да ступи са њим у комуникацију уколико се за тим укаже потреба.

Пројекат на коме ради уловани запослени и списак колега коју су на том пројекту је истакнут другом бојом. Такође је су и информације улованог запосленог графички другачије представљене. Тиме се постиже да буду наглашени подаци који су најзначајнији за корисника, а то су свакако пројекат на којем ради и колеге задужене за тај пројекат. С друге стране, уколико му је потребан и неко са другог задужења или пројекта, ова функционалност му омогућава интуитиван увид и у то.

#### *ii) Приказ сопственог календара и резервација радног места*

Наредни кориснички захтев за запосленог је приказ његовог распореда долазака у просторије фирме у виду календара. У оквиру ове функционалности реализована је и могућност резервације радног места, одн. отказивања истог.

Запосленом је приказан календар на месечном нивоу, који је подразумевано постављен на текући месец. Селектовањем жељеног датума, учитава се списак свих запослених, груписаних по пројектима, коју су изабрали да тај дан раде из фирме. За сваки датум се приказује и број преосталих слободних радних места.

Датуми за које се запослени одлучио да ради из канцеларије су визуелно истакнути. Ако запослени кликне на неки од њих, доступна му је опција отказивања доласка у фирму. Међутим, уколико селекује неки од необележених датума, имаће могућност да се пријави да тај радни дан ради из просторија фирме. Ако у том тренутку има слободно место, датум ће бити увршћен у распоред долазака запосленог. У супротном ће захтев бити послат серверској страни. После обраде ће бити приказан администратору кроз поменути статистику попуњености. Колонa која представља број запослених који тог радног дана нису или места биће увећана за један.

#### *iii) Преглед распореда долазака изабраног колеге*

На истој страници на којој се налази кориснички захтев прегледа сопственог календара, налази се и функционалност приказа распореда долазака изабраног колеге. Током имплементације система одлучено је да ове две опције буду реализоване на истој страници. На овај начин се омогућава да уловани запослени може да ажурира свој распоред у складу са својим колегама.

И ова функционалност, као и претходна, реализована је кроз календар. Наиме, запослени у назначено поље треба да унесе колегу чији распоред долазака жели да погледа. Потом се учитава календар траженог колеге. Дани које је колега изабрао као дане доласка биће визуелно назначени одговарајућом бојом.

#### iv) Приказ распореда седења

Кориснички захтев приказа распореда седења запосленима представља једну од главних функционалности овог система. Пошто историјат распореда седења није релевантан за корисника, да би систем био интуитивнији и ослобођен сувишних детаља, запосленима се приказује распоред седења за текући одн. наредни радни дан.

Уколико се систему приступа пре 20h, улогованом запосленом ће се приказати распоред седења за текући радни дан. Међутим, ако запослени приступа апликацији након поменутог времена, распоред текућег дана није више релевантан ни актуелан. Штавише, алгоритам је извршио генерисање распореда седења запослених за наредни радни дан. Стога ће након 20h свим логованим запосленима бити доступан распоред за наредни дан.

И у случају приказа распореда седења за текући, и у случају распореда седења за наредни радни дан, кориснику се првенствено истиче у коју канцеларију га је алгоритам сврстао и на ком спрату се она налази. Уколико корисник није попунио пријаву доласка за тај дан, исписује му се одговарајућа порука.

Сви корисници имају приступ прегледу распореда седења, и у случају да су се пријавили за тај дан, и ако нису. Избором канцеларије учитава се списак запослених којима је додељена та канцеларија. Такође постоји и информација колико је слободних места преостало. На овај начин, уколико запослени није резервисао место за долазак, може и накнадно да се одлучи да текући дан ради из фирме.

### 3.2. Опис коришћених технологија

У наставку је дат осврт на технологије које су коришћене у изради овог система. Представљене су и библиотеке и платформе захваљујући којима су олакшане реализације појединих функционалности.

#### 3.2.1. Клијентска страна

За реализацију клијентске стране коришћена је *React* библиотека [5] (верзија 18.2.0). *React* представља *JavaScript* библиотеку отвореног кода која се користи за изградњу корисничких интерфејса. Техника приступа која се примењује је *SPA (Single Page Application)*. Она омогућава да корисник током целог коришћења система ниједном не освежи страницу, одн. да се садржај једине странице динамички мења пратећи рад корисника.

Током имплементације појединих функционалности на клијентској страни коришћени су различите *React* библиотеке и алати. Ефикаснији развој веб форми и напредних веб компоненти имплементиран је уз помоћ *Bootstrap* оквира. То је библиотека отвореног кода који представља комбинацију *HTML*-а, *CSS*-а и *JavaScript*-а. Обезбеђује све основне компоненте потребне за развој потпуно функционалног веб корисничког интерфејса. Модуларног је типа те може бити лако надограђиван.

За управљање рутама у веб апликацији коришћен је *React Router*. То је библиотека која се користи за динамичко рутирање и управља процесом навигације у веб апликацији. Динамичко рутирање се одвија док се апликација извршава на клијентској машини. Омогућава навигацију између различитих компоненти у *React* апликацији, дозвољава промену *URL*-а претраживача и одржава синхронизацију корисничког интерфејса са *URL*-ом.

Да би одређене информације биле приказале кориснику коришћена је *React-Toastify* библиотека. Она омогућава да се једноставно прикажу обавештења која су релевантна за корисника. Обавештења искачу као поруке које приказују одређене информације кориснику. Обавештења могу бити поруке о успеху, упозорењу, грешкама. Коришћена су свуда у апликацији где је било потребе приказати кориснику неку додатну информацију.

За статистичку анализу попуњености канцеларија по радним недељама и исцртавање одговарајућих графика коришћена је библиотека *react-chartjs-2*. Она представља *React* омотач за *Chart.js*. *Chart.js* је *npm* пакет који омогућава креирање и исцртавање графика и графика.

### 3.2.2. Серверска страна

Серверски део апликације имплементиран је уз помоћ *Node.js* окружења [6] (верзија 16.14.2). *Node.js* је вишеплатформско окружење намењено за извршавање *JavaScript* кода на серверској страни. За ефикасније организовање програмског кода и имплементирање логике на серверу коришћен је програмски оквир *Express* [7] (верзија 4.18.1). Он иницијализује сервер, омогућава рад са *http* захтевима који ту пристижу, обрађује их и враћа *http* одговор клијентској страни.

#### i) *Node.js* модули

*Node.js* се састоји од модула који се укључују у пројекат у зависности од потребе. Поред *Express*-а, модули који су употребљени у овој апликацији су и *body-parser* и *cors*. Да би се добио приступ подацима из захтева неопходно је користити *body-parser*. Он омогућава да *Express* прочита тело захтева и да га парсира у *JSON* (*JavaScript Object Notation*) објекат који му је разумљив. С друге стране, *cors* је *Node.js* пакет који обезбеђује *Connect/Express* посреднички софтвер и може омогућити *CORS* (*Cross-Origin Resource Sharing*) са различитим опцијама. *CORS* је механизам који подржава сигурне захтеве са више извора и пренос података између претраживача и сервера.

#### ii) *HTTP / HTTPS* протокол

*HTTP / HTTPS* (*Hypertext Transfer Protocol / Secure*) је најчешћа метода преноса информација на интернету, с разликом да се код *HTTPS* протокола креира безбедни канал преко незаштићене мреже. Ове методе служе за комуникацију између сервера и клијента. Оне функционишу по принципу *захтев / одговор*. Основна намена ових протокола је приказивање *HTML* докумената, одн. интернет страница. *HTTP* сервер константно ослушкује захтеве на одређеном мрежном комуникацијском порту, чекајући да се клијент повеже и пошаље захтев. Након што *HTTP* сервер одговори на захтев клијента, конекција се прекида до следећег захтева.

Модули *http* и *https* су једни од најважнијих модула које чине језгро *Node.js* технологије. У пракси се за реализацију серверске стране најчешће користи поменути *Express* модул. Он се базира на *http* и *https* модулима, али омогућава више функционалности и чини код добро организованим. Након што се сервер креира, поставе се ослушкивачи на одређени порт, и тиме се омогућава прихватање и обрада тих захтева.

#### iii) *REST* архитектура

*REST* (*REpresentational State Transfer*) је архитектура коришћена за израду серверске стране апликације. Заснована је на веб стандардима и користи *HTTP* протокол. Захтеви који се шаљу на сервер поред заглавља морају садржати и тип. Постоје четири најчешћа типа захтева, а то су:

- *GET* – захтев да сервер пошаље одређени податак из базе
- *POST* - захтев да сервер упише у базу нов податак који се налази у захтеву
- *PUT* – захтев да сервер измени одређени податак у бази
- *DELETE* – захтев да сервер обрише одређени податак из базе

#### iv) Алгоритам расподеле

Алгоритам расподеле запослених реализован је на серверској страни у програмском језику *Python* у виду скрипте. Идеја је да се алгоритам извршава сваког радног дана у 20h и да генерише распоред седења за наредни радни дан. Да би се омогућило извршавање скрипте у одређеном тренутку коришћен је *node-cron* модул [8] (верзија 3.0.1). Он представља планер задатака у *JavaScript*-у за *Node.js* и омогућава њихово заказивање и извршавање у жељено време.

### 3.2.3. База података

За трајно складиштење података коришћена је *MongoDB* нерелациона документно оријентисана база података [9] (верзија 5.0.8). Базу података код *MongoDB*-а сачињавају колекције докумената. Документ је скуп парова облика кључ – вредност, док је колекција скуп докумената. Колекција је еквивалентна табели код релационих база података. Разлика је у томе што не морају сви документи унутар једне колекције да имају иста поља, тј. документи могу имати различите шеме.

*MongoDB* складишти податке у објектима који су *BSON* (*Binary JavaScript Object Notation*) облика. *BSON* проширује *JSON* репрезентацију објеката на додатне врсте као што су *int*, *long*, *date*, *floating point*, *decimal128*.

*MongoDB* има званичне драјвере за различите програмске језике. За потребе пројекта описаног овим документом коришћен је *Mongoose* [10] (верзија 5.0.8). *Mongoose* је *JavaScript* објектно-оријентисана програмска библиотека која креира конекцију између *MongoDB* базе података и *Node.js* окружења. Она омогућава лакше руковање базом, обезбеђује валидацију шеме, управља односима између података, преводи објекте у коду и представља их у *MongoDB* бази.

Да би алгоритам расподеле запослених реализован у *Python* скрипти могао да се повеже са *MongoDB* базом података коришћен је *PyMongo* драјвер [11]. Он омогућава рад са подацима из базе, и представља препоручени начин рада са *MongoDB* базом података у *Python*-у.

### 3.2.4. Платформе за извршавање апликације

Једна од платформи коришћена за реализацију овог система је *Firebase* платформа [12]. Она је производ компаније *Google*. Пружа услуге базе података за рад у реалном времену и *backend* услуге. За потребе овог пројекта коришћена је *Firebase* аутентикација. То је сервис који подржава аутентикацију корисника и омогућава персонализовано искуство и безбедно чување корисничких података. У апликацији су конкретно логовање и регистрација путем имејл адресе омогућене захваљујући овом сервису.

Наредна платформа примењена у изградњи овог система је *EmailJS* платформа [13]. Она омогућава слање и пријем имејлова без коришћења серверске стране. Има интуитиван кориснички интерфејс који омогућава једноставно креирање шаблона е-поште са динамичким параметрима. У пројекту је коришћена за потребе позивања запослених у

апликацију од стране администратора. Такође је употребљена и за комуникацију запосленог и администраторске стране у случају да неким превидом запосленом није додељена одговарајућа компанија.



## 4. ОПИС РАДА СИСТЕМА

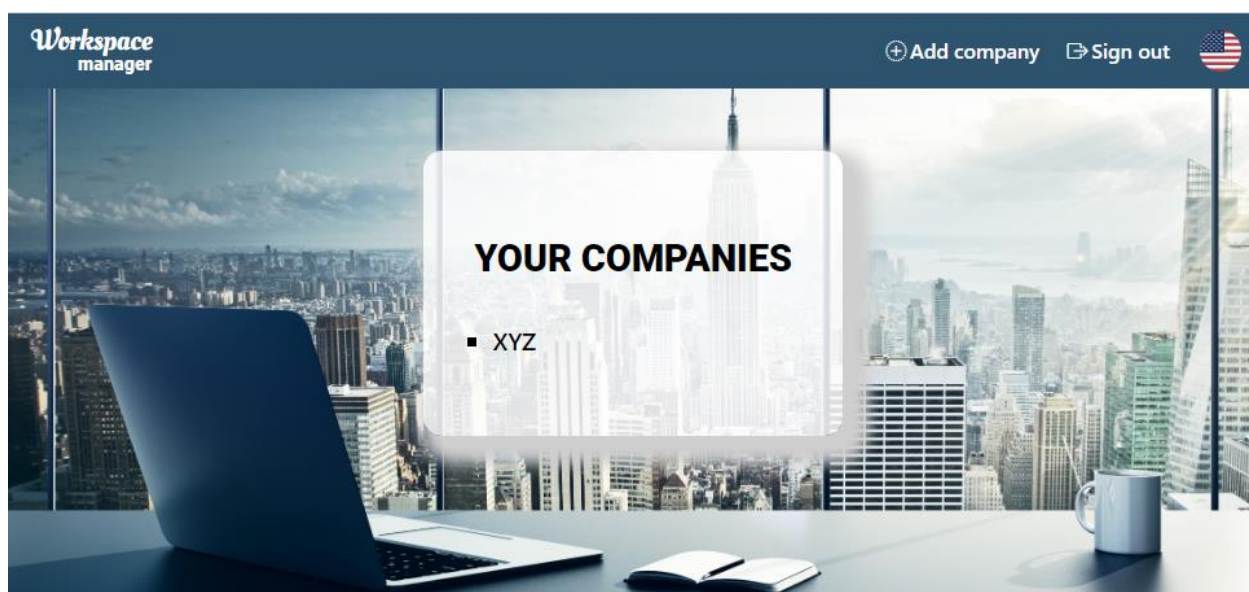
У овом поглављу дат је преглед главних функционалности система приказаних кроз корисничке екране. Како постоје два типа корисника – администратор и запослени, то су функционалности надаље груписане и представљене по типу корисника. Оно што је заједничко за све кориснике су свакако странице за логовање и регистрацију.

### 4.1. Кориснички екрани администратора

Надаље ће бити приказани кориснички екрани који одговарају администраторском типу корисника.

#### 4.1.1. Главни кориснички екран администратора

Након успешне пријаве у систем, администратору се приказује главни кориснички екран дат на слици 4.1.1.1. На тој страници се изLISTАВАЈУ све компаније које је креирао администратор и којима управља. Дугме *Add Company* на навигационом бару одводи администратора на страницу на којој уношењем одговарајућих података може да креира нову компанију.



Слика 4.1.1.1. Главни кориснички екран администратора

Кликом на назив једне од компанија администратор се одводи на страницу која садржи детаље компаније. На тој страници налазе се информације као што су адреса, контакт имејл адреса, контакт телефон и радно време. Администратор има могућност брисања и ажутирања компаније. Навигациони бар ове странице садржи линкове ка канцеларијама, пројектима, запосленима, распореду седења и анализи попуњености изабране компаније.

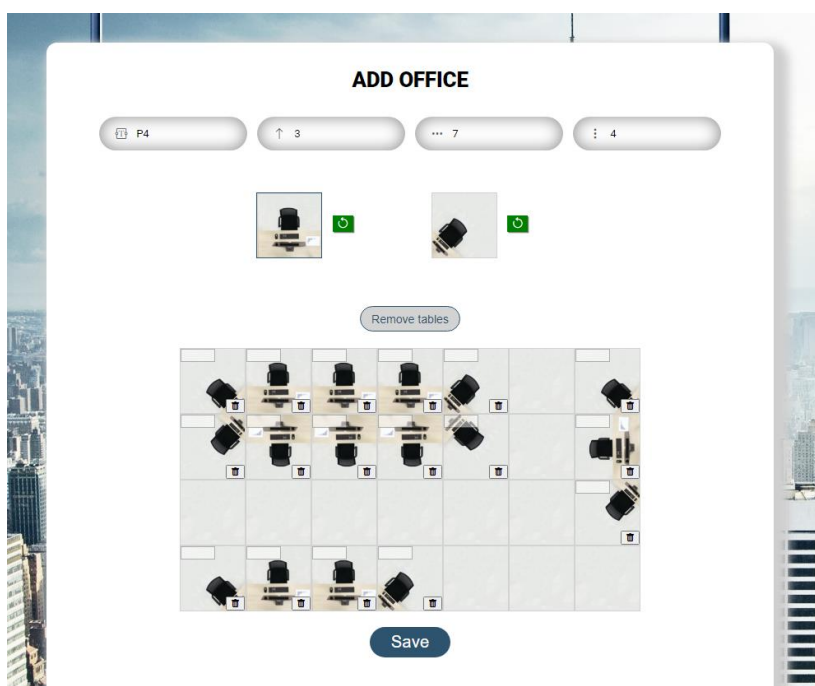
#### 4.1.2. Канцеларије

Кликом на линк *Offices* навигационог бара одлази се на страницу која садржи канцеларије груписане по спратовима, која је приказана на слици 4.1.2.1. Овде постоји могућност креирања нове канцеларије, као и прегледа постојеће. Кликом на изабрану канцеларију одлази се на страницу на којој се налазе њени детаљи. Ту су такође имплементирани и опције брисања и ажурирања канцеларије.



Слика 4.1.2.1. Приказ свих канцеларија

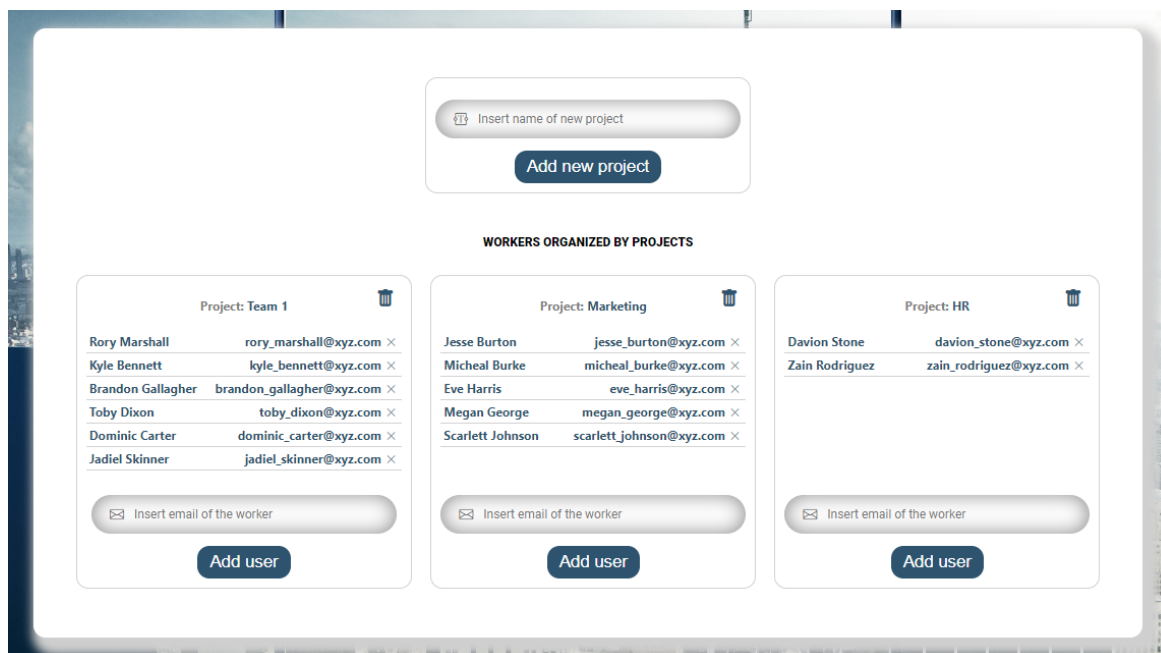
На слици 4.1.2.2. приказана је страница за креирање канцеларије. Код креирања уноси се њен назив, спрат на којем се налази, као и колико столова би могло да стане по дужини и ширини. Потом се превлачењем дугмића са сликом радног места прави тлоцрт канцеларије. Шема канцеларије може да се попуњава и обичним кликом на изабрано место.



Слика 4.1.2.2. Креирање канцеларије

### 4.1.3. Пројекти

На слици 4.1.3.1. приказан је део странице са детаљима о пројектима. На врху се налази форма за креирање новог пројекта. Потом су приказани креирани пројекти у виду листе картица, где се за сваки пројекат налази и списак запослених који раде на њему. Могуће је брисање пројекта, као и ажурирање запослених на пројекту кроз функције додавања и брисања запослених.



Слика 4.1.3.1. Страница са детаљима о пројектима

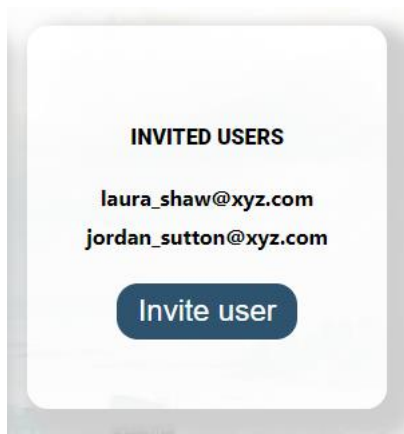
### 4.1.4. Запослени

На слици 4.1.4.1. приказан је део странице на коју се одлази када се из навигационог бара изабере линк *Workers*. Ту се налази списак свих запослених са информацијама о њиховим именима, имејл адреси, и пројекту или задужењу за које је запослени овлашћен.



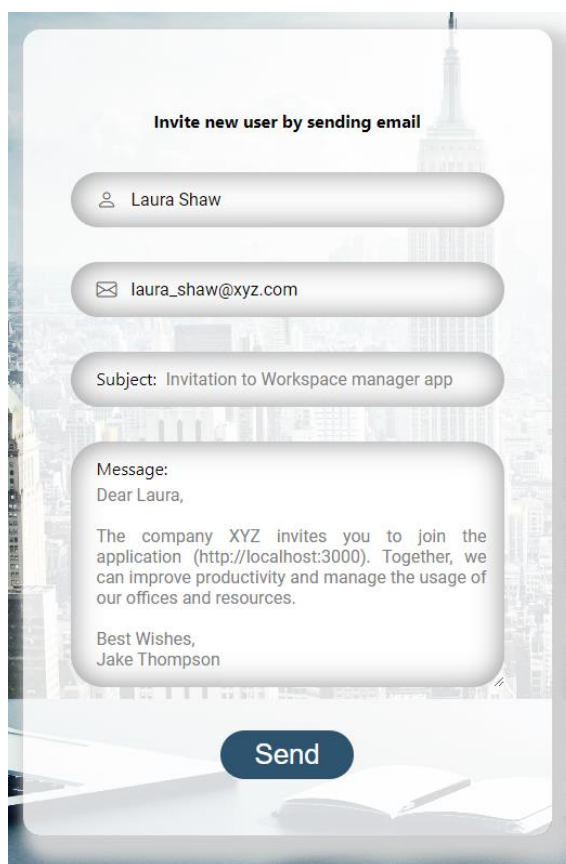
Слика 4.1.4.1. Списак запослених

На слици 4.1.4.2. приказан је десни део исте странице у виду картице. Овде се налази списак свих запослених који су позвани у апликацију од стране администратора, а нису се још регистровали. Једном када се региструју биће приказани заједно са осталим колегама у поменутој секцији регистрованих запослених.



Слика 4.1.4.2. Корисници позвани у апликацију

У доњем делу поменуте картице налази се и дугме које одводи на страницу чији је део приказан на слици 4.1.4.3. Са ове странице администратор може да позове запосленог да се придружи апликацији уношењем његовог имена и имејл адресе. Наслов имејла као и текст поруке се аутоматски генеришу. Међутим, како се ради о пољима за унос, администратор може и да их измени уколико му то више одговара.



Слика 4.1.4.3. Позивање запосленог путем имејла у апликацију

#### 4.1.5. Распоред седења

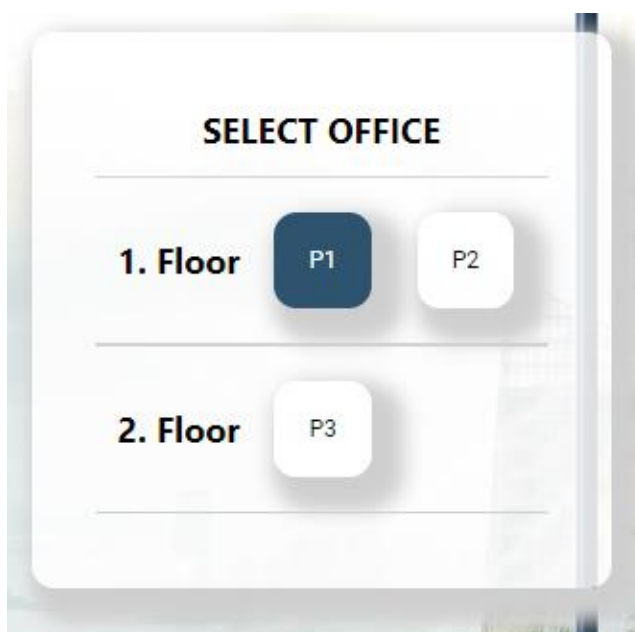
Кликом на линк *Seating Chart* навигационог бара администратор се преусмерава на страницу са распоредом седења запослених по датумима. На слици 4.1.5.1. приказан је леви горњи угао те странице. Ту се кликом на иконицу календара може селектовати датум за који администратор жели да погледа распоред седења.



Слика 4.1.5.1. Избор датума

Апликација ради по принципу да су пријаве запослених за долазак у канцеларију за сутрашњи дан отворене до неког предефинисаног времена. Узето је да је то време 20h текућег дана. После тога пријаве се затварају и извршава се алгоритам који генерише распоред седења. Дакле, после 20h биће доступан и распоред седења за сутрашњи дан, а пре тог времена само за данашњи. У складу с тим, на овој страници је доступан историјат седења, као и распоред за данашњи одн. сутрашњи дан у зависности да ли су отворене или затворене пријаве за долазак. Уколико се изабере неки датум у будућности, исписује се порука да распоред седења није још генерисан.

На слици 4.1.5.2. приказан је горњи десни угао поменуте странице. Ту се налазе имена свих канцеларија груписана по спратовима. Кликом на једну од њих приказује се списак запослених које је алгоритам одредио да буду у изабраној канцеларији. Селектована канцеларија има картицу плаве боје као што је и приказано на слици.



Слика 4.1.5.2. Избор канцеларије и приказ распореда седења за одабрану канцеларију



На слици 4.1.5.3. приказан је списак запослених који је алгоритам генерисао за изабрани датум и изабрану канцеларију. Алгоритам ради по принципу да људи из истог тима буду распоређени у истој канцеларији ако је то физички могуће. Поред имена и имејл адреса запослених налазе се и информације ко припада ком тиму. Такође је и у горњем десном углу назначено колико је остало слободних места у канцеларији за тај дан.

Number of free seats in office: 5	
<b>Brandon Gallagher</b> brandon_gallagher@xyz.com Project: Team 11	<b>Micheal Burke</b> micheal_burke@xyz.com Project: Marketing
<b>Toby Dixon</b> toby_dixon@xyz.com Project: Team 11	<b>Eve Harris</b> eve_harris@xyz.com Project: Marketing
<b>Dominic Carter</b> dominic_carter@xyz.com Project: Team 11	<b>Megan George</b> megan_george@xyz.com Project: Marketing
<b>Jadiel Skinner</b> jadiel_skinner@xyz.com Project: Team 11	<b>Davion Stone</b> davion_stone@xyz.com Project: HR
<b>Jesse Burton</b> jesse_burton@xyz.com Project: Marketing	<b>Zain Rodriguez</b> zain_rodriguez@xyz.com Project: HR

Слика 4.1.5.3. Списак запослених у изабраној канцеларији за изабрани датум

На слици 4.1.5.4. је приказан део исте странице. Ту је графички исцртана визуелна репрезентација столова у селектованој канцеларији.



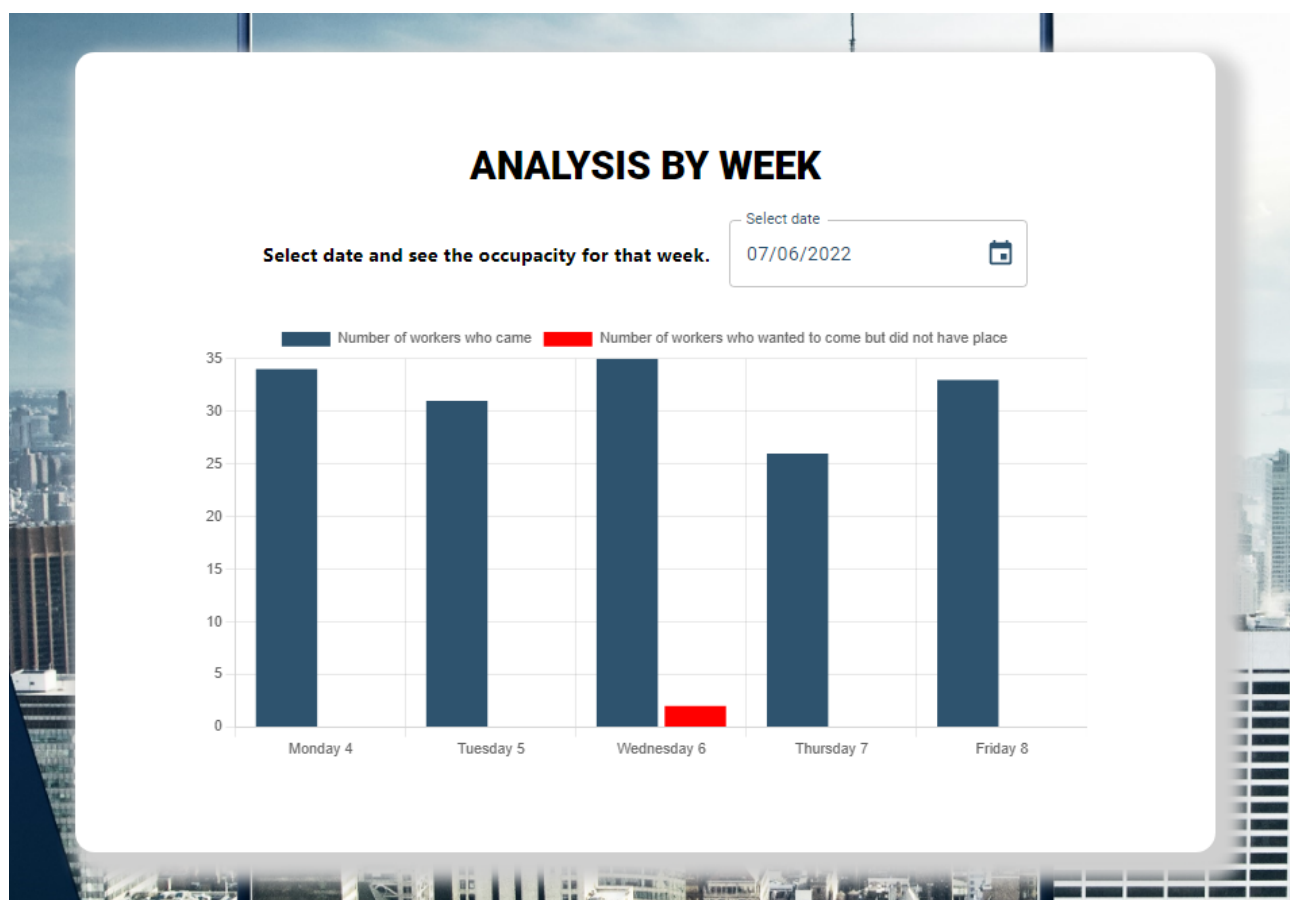
Слика 4.1.5.4. Тлоцрт канцеларије

#### 4.1.6. Статистика попуњености

На слици 4.1.6.1. приказан је део странице на којој се одлази када администратор кликне на линк *Analysis* навигационог бара. Ту се налази реализација важне функционалности за аналитику попуњености канцеларија у претходном периоду. На овај начин администратор има увид у број искоришћених радних места на дневном и недељном нивоу. На основу ове функционалности управа компаније може одлучивати о евентуалном ажурирању капацитета компаније у зависности од навика и жеља запослених.

На врху странице се кликом на иконицу календара може изабрати радна недеља за којој администратор жели да погледа искоришћеност канцеларија. Потом се учитава статистика попуњености у виду хистограма за изабрану недељу. На страници су приказане и додатне лабеле да би читање графикана било интуитивно.

Хистограм садржи две врсте колона за сваки радни дан у недељи. Прва колона приказује колико је радних места заузето од укупног броја места у фирми. Друга приказује колико је било запослених који су хтели да дођу у канцеларију одређеног дана, али није остало слободних места за њих. Зато је ова колона и приказана црвеном бојом, да би истакло администратору да тог дана није било довољно места. Уколико је ово тренд који се понавља, управи компаније ће се јасно ставити до знања да треба да повећа капацитете фирме. С друге стране, уколико хистограм показује да попуњеност никад не прелази одређени проценат, то указује да капацитети фирме могу да се смање.



Слика 4.1.6.1. Анализа попуњености

## 4.2. Кориснички екрани запосленог

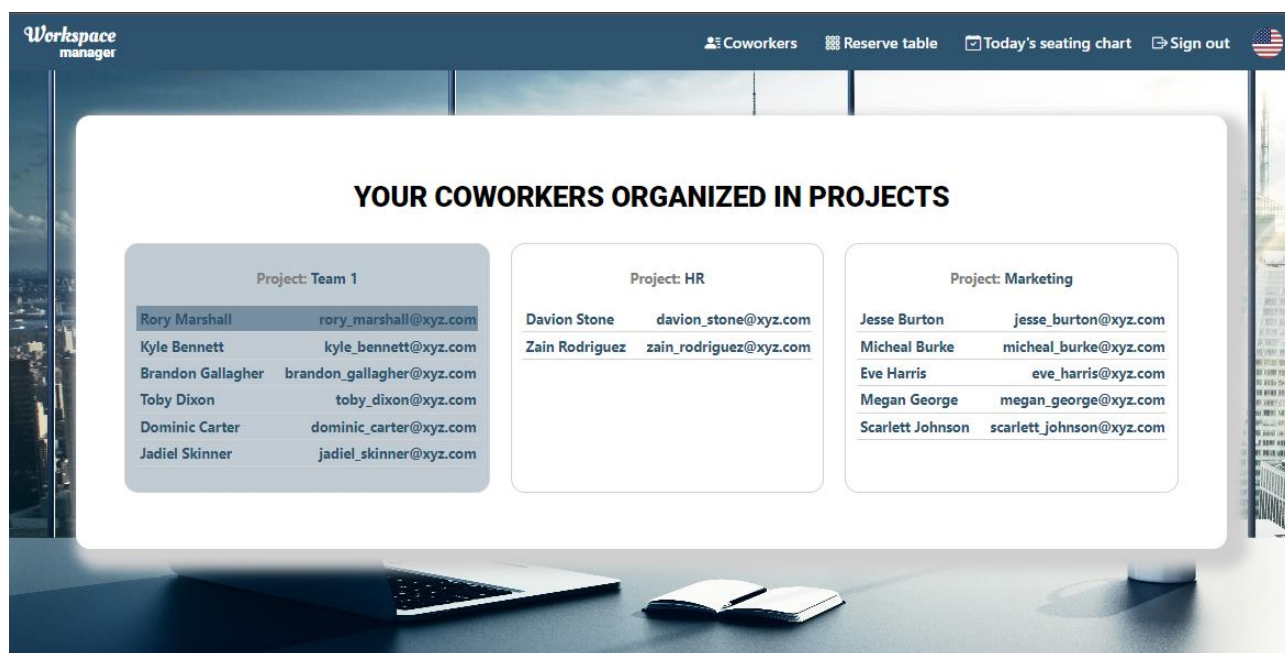
Надаље ће бити приказани странице на које се одлази после успешног логовања или регистрација запосленог. Биће представљене три главне функционалности, а то су: преглед колега логованог запосленог, резервација радног места и приказ распореда седења.

### 4.2.1. Колеге и њихова организација по пројектима

На слици 4.2.1.1. приказана је страница улогованог запосленог на коју се одлази када се кликне линк *Coworkers*. Ту се налази списак свих пројеката и задужења у оквиру фирме који су графички представљени у виду картица. У оквиру картице налази се и списак свих запослених који су задужени за ту активност, као и њихове имејл адресе.

Ангажовање логованог запосленог биће истакнуто светло плавом бојом као на слици. Такође ће и његово име бити назначено нешто тамнијом бојом. Тако се постиже бољи визуелни ефекат и релевантни подаци су уочљивији.

Улогованом запосленом је најбитније да види своје задужење, као и сараднике којима је додељена иста активност. С друге стране, постоји и могућност да прегледа и остале активности фирме, као и да види које колеге су на тим задужењима. Уколико су му потребне колеге са одређене активности, прегледом ове странице ће их ефикасно наћи.



Слика 4.2.1.1. Страница са детаљима о пројектима и колегама

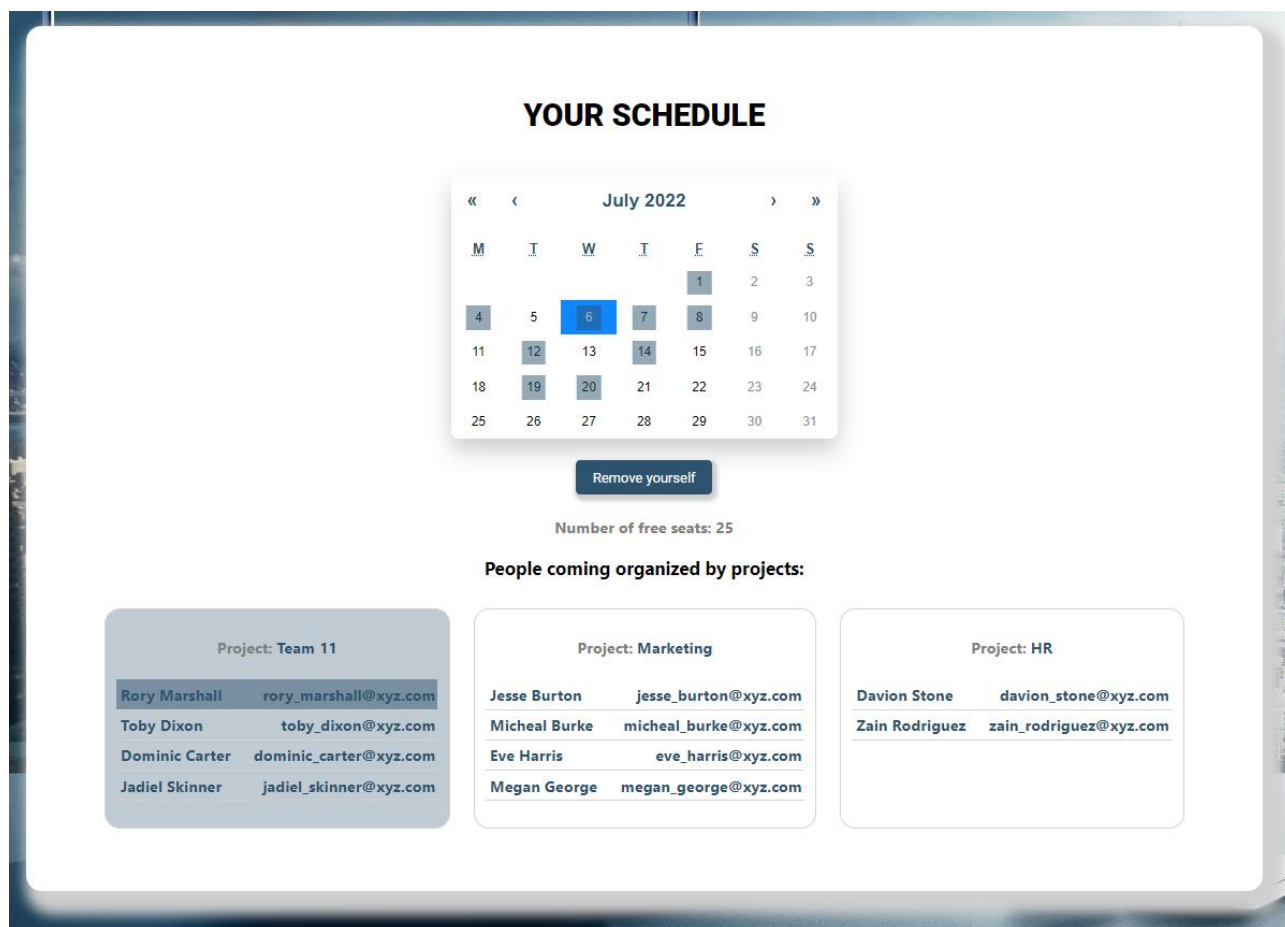
### 4.2.2. Резервација радног места

На слици 4.2.2.1. приказан је део странице на који се одлази када запослени кликне линк *Reserve table* навигационог бара. Главна функционалност ове странице је да запослени има визуелну репрезентацију оних дана које је изабрао да ради из канцеларије. Додатно, реализована је и функционалност да запослени може да види и календаре својих колега, и да у складу с тим ажурира свој.

На самом врху странице налази се календар на месечном нивоу који је подразумевано постављен на текући месец. Сивом бојом су обојени и уоквирени они дани за које се



запослени већ одлучио да ради из канцеларије. Дан који запослени кликом изабере да би видео његове додатне информације је истакнут плавом бојом, док су преостали дани без визуелне назнаке.



Слика 4.2.2.1. Резервација радног места

Уколико запослени кликне на дан који је већ изабран за долазак, на страници ће постојати дугме *Remove yourself*. Притиском на то дугме запослени отказује долазак, а селектовани датум престаје да буде сиве боје.

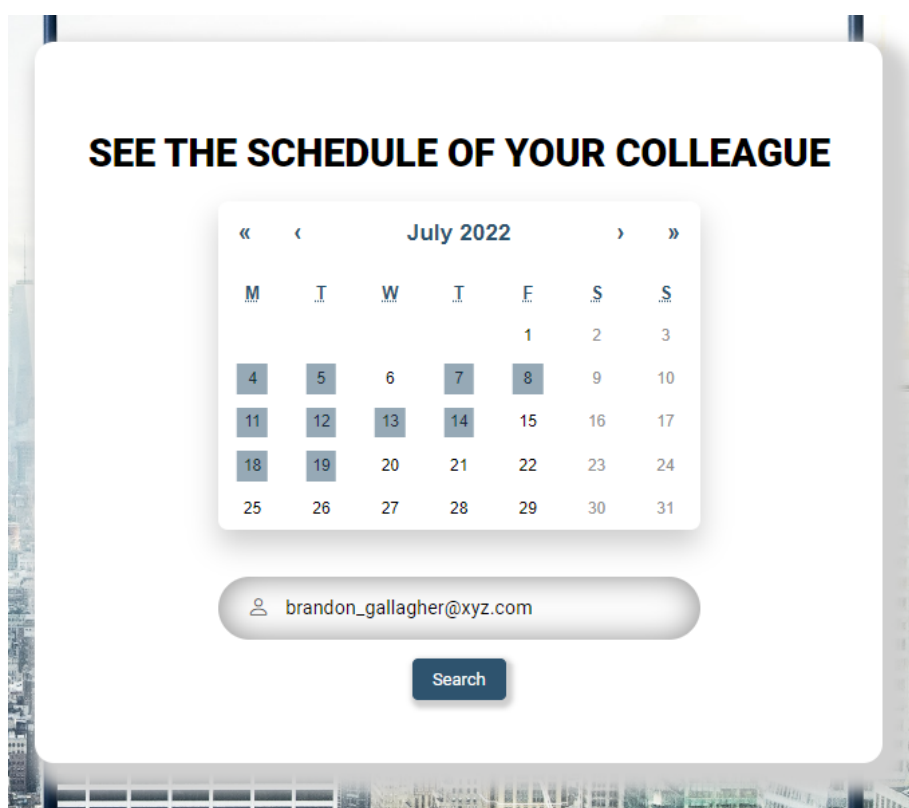
С друге стране, уколико запослени селекује дан који нема никакве визуелне ознаке одн. дан за који се није до тада одлучио да ради из канцеларије, у додатним информацијама тог дана налазиће се и дугме *Add yourself*. Кликком на то дугме запосленом се додаје изабрани датум међу дане за које се определио да ради из канцеларије, и селектовани датум постаје сиве боје. Међутим, уколико су сва места попуњена за тај дан, кориснику се исписује одговарајућа порука. У том случају на серверу се инкрементира број запослених који тог дана нису имали радно место у фирми. Та информација биће визуелно представљена у поменутој статистичкој анализи администратора, као црвена колона хистограма, висине једнаке броју запослених који нису имали радно место тог дана.

Оно што је заједничко за додатне информације сваког од датума је да увек исписана вредност бројача преосталих слободних места. Између осталог, постоји и списак запослених који су резервисали радно место за изабрани датум. Списак запослених је и овде организован по пројектима одн. задужењима. Активност уложеног корисника за селектовани дан биће истакнута сивом бојом, уколико постоји бар један запослени са тог задужења који је изабрао

дати дан. У супротном, ниједна активност неће бити посебно назначена. Такође, уколико је улоговани запослени кликнуо на дан који је међу његовим данима за долазак, у списку запослених ће његове име бити истакнуто тамнијом бојом, као што је и приказано на слици.

С друге стране, уколико запослени жели да се пријави за долазак у канцеларију за наредни дан, а то уради после 20h текућег дана, биће приказана одговарајућа порука. Наиме, као што је више пута речено, алгоритам расподеле запослених по канцеларијама за сутрашњи датум се извршава у 20h текућег дана. Дакле, после тог времена распоред је већ генерисан, а пријаве за наредни дан ће бити затворене.

На слици 4.2.2.2. приказана је наредна функционалност исте странице. Визуелна репрезентација претходно описане имплементације одговара левој половини екрана. Функционалност која ће бити описана у наставку налази се у десном делу поменутог странице.



Слика 4.2.2.2. Претрага распореда долазака од колега

Овај део екрана представља репрезентацију календара од колега улогованог запосленог. Наиме, испод календара налази се поље за унос са подразумеваном поруком да се унесе имејл адреса колеге чији распоред долазака жели да се погледа. Уколико се унесе непостојећа имејл адреса, кориснику ће бити приказана одговарајућа порука. У супротном се учитава календар траженог колега постављен на текући месец, с могућношћу претраге и осталих месеци.

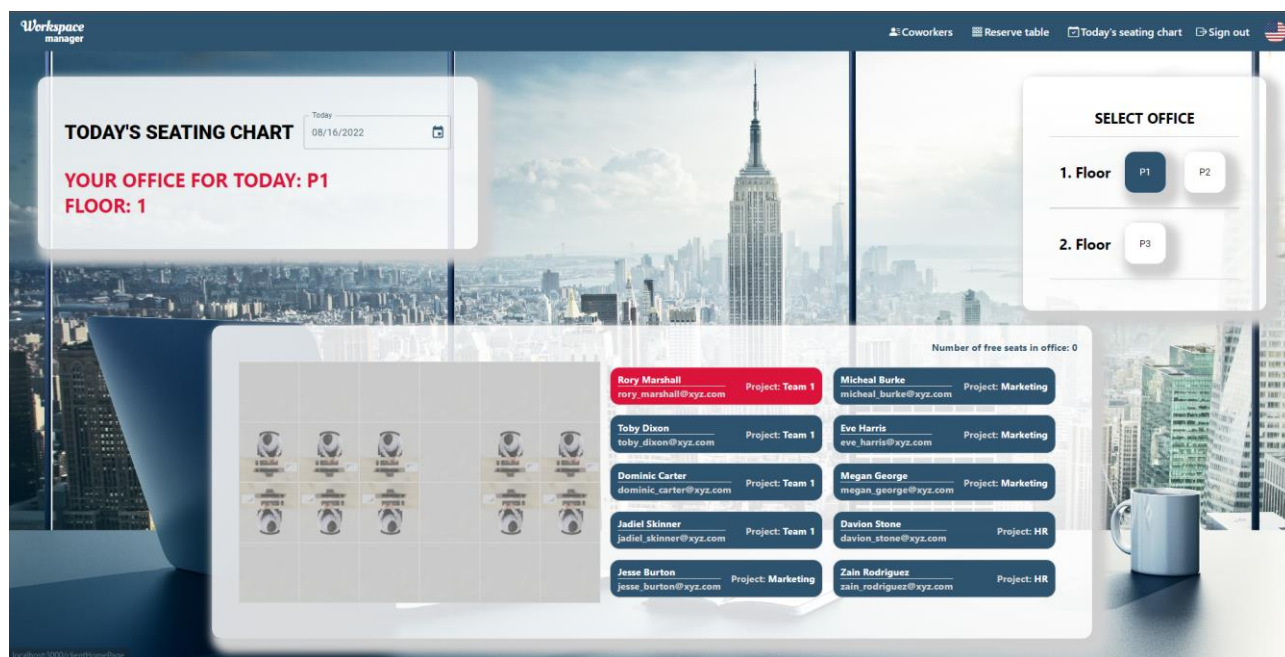
Као и код прегледа сопственог календара, тако су и код представљања календара од колега дани доласка репрезентовани сивом бојом, док остали немају никакво обележје. Како су и календар улогованог запосленог и календар изабраног колега на истој страници, то је поједностављено и ажурирање сопственог распореда долазака са распоредом колега.

### 4.2.3. Приказ распореда седења по канцеларијама

На слици 4.2.3.1. приказана је страница на којој се преусмерава логовани запослени када кликне на трећи линк навигационог бара. На овој страници је имплементиран визуелни приказ резултата алгорита који распоређује запослене по канцеларијама.

За разлику од администраторске функционалности прегледа распореда седења по датумима, за запослене није релевантан приказ распореда за датуме који су прошли. Такође, после краја радног времена престаје да буде значајно ко је седео у којој канцеларији. Стога, да би апликација била интуитивнија и без сувишних детаља, запослени имају приступ распореду седења за текући дан, одн. за наредни радни дан, у зависности од тога да ли се претражује распоред седења пре, током и после краја радног времена.

Алгоритам расподеле запослених такав да запослени из истог тима одн. са истим задужењем буду у истој канцеларији се извршава сваког радног дана у 20h, и он генерише распоред за наредни радни дан. У складу с тим, трећи линк навигационог бара носиће назив *Today's seating chart* уколико се апликацији приступа пре предефинисаног времена, и биће приказан распоред за текући радни дан. Уколико се апликацији приступа после 20h, трећи линк навигационог бара зваће се *Tomorrow's seating chart* и биће приказан распоред седења за наредни радни дан.



Слика 4.2.3.1. Страница са приказом распореда седења по канцеларијама

У левом горњем углу поменуте странице налази се картица са основним информацијама везаним за распоред седења улогованог запосленог. На врху те картице налази се одговарајући текст у зависности да ли је приказан распоред текућег или наредног радног дана. У наставку исте хоризонтале је приказан и тачан датум на који се односи распоред седења. Испод тога се налазе информације у виду текста које означавају где је запослени распоређен. Уколико није попунио пријаву за долазак за означени датум биће приказана одговарајућа порука. У супротном биће исписано која канцеларија му је додељена и на ком спрату фирме се налази.

У горњем десном углу исте странице налази се картица која репрезентује канцеларије фирме организоване по спратовима. Селектована канцеларија биће приказана тамно плавом бојом. Уколико се улоговани корисник налази у распореду за седење онда ће канцеларија која му је додељена бити иницијално селектована. У супротном ће после учитавања странице бити означена прва канцеларија на најнижем спрату.

У централном делу странице биће приказани детаљи распореда запослених за селектовану канцеларију. У левом делу се налази графички приказ распореда радних места у одабраној канцеларији који омогућава и визуелно разликовање канцеларија. У десном делу поменуте секције налази се списак људи којима је додељена селектована канцеларија. Информације улогованог запосленог биће представљене црвеном бојом ради лакше уочљивости. За сваког запосленог приказано је његово име и презиме, имејл адреса, и пројекат или задужење које му је додељено. На овај начин се постиже и визуелна представа резултата рада алгоритма. Штавише, у горњем десном углу налази се и информација колико слободних места је остало за селектовану канцеларију. Стога, уколико се запослени није пријавио на време путем анкете, у овој секцији може видети да ли постоји негде слободно место, и у складу с тим накнадно одлучити да ли ће тај дан радити удаљено или доћи у канцеларију.

## 5. РЕАЛИЗАЦИЈА СИСТЕМА

У овом поглављу представљена је реализација кључних корисничких захтева подржаних у систему. Стављен је фокус на изазове који су се појавили приликом израде саме апликације. У наставку текста приказан је начин на који су имплементиране најважније функционалности, како на клијентској тако и на серверској страни.

### 5.1. Реализација серверске стране

У овој секцији приказани су најбитнији делови који су омогућили реализацију серверске стране апликације. Надаље су представљени модели података који су коришћени, начин на који је сервер конфигурисан, прихватање захтева на серверу, као и идејно решење алгоритма расподеле запослених.

#### 5.1.1. Модели података

Модели података омогућавају ефикасан рад са подацима који се чувају у бази. То су конструктори вишег реда који узимају шему и креирају инстанцу документа. Они олакшавају управљање подацима и чине код добро организованим. На слици 5.1.1.1. представљени су сви модели који су коришћени за рад са подацима у бази овог система.

```
const User = mongoose.model("user", userSchema, "user");
const Company = mongoose.model("company", companySchema, "company");
const UserCompany = mongoose.model("user_company", userCompanySchema, "user_company");
const Office = mongoose.model("office", officeSchema, "office");
const Project = mongoose.model("project", projectSchema, "project");
const Survey = mongoose.model("survey", surveySchema, "survey");
const Reservation = mongoose.model("reservation", reservationSchema, "reservation");
```

Слика 5.1.1.1. Модели података

Као што се види са слике, у систему постоји пре свега модел *User* који репрезентује корисника. Овде је од значаја поље *admin* које је *Boolean* типа. На основу њега се разликује да ли је корисник администратор или запослени. Даље имамо модел података под називом *Company* који представља компанију и све информације које су од значаја за њу. Модел података *UserCompany* представља везу између корисника и компаније.

Модели *Office*, *Project*, *Survey*, *Reservation* имају поље *company\_id*. Оно носи информацију о којој компанији се ради када се посматра одређени модел. Модел *Office* садржи све информације које су релевантне за канцеларију компаније. *Project* представља пројекат који је администратор креирао, и садржи листу имејл адреса свих запослених који раде на том пројекту.

*Survey* је репрезентација анкете у којој се запослени изјашњавају за датуме када желе да раде из просторија компаније. Овај модел је реализован тако да садржи поље које представља датум за долазак у канцеларију. Ту се налази и списак свих запослених који су се

изјаснили да би радили тај дан из фирме. Такође, постоји и поље које представља списак оних који су се пријавили за поменути датум, али су сва места у том тренутку већ била попуњена.

*Reservation* је модел података који представља резултат рада алгоритма. Садржи поља *date* и *offices*. *Date* репрезентује датум на који се односи генерисани распоред седења *Offices* се састоји од низа објеката са пољима (*office\_id*, *users*). *Office\_id* означава везу са канцеларијом на коју се распоред односи. Поље *users* представља списак запослених којима је додељена поменути канцеларија за тај датум.

### 5.1.2. Дефинисање рута

Систем је реализован тако да свака рута буде задужена за један модел података. Она обрађује све захтеве који пристижу на сервер, а везани су за тај модел. На слици 5.1.2.1. је приказана рута *userRoutes*. Она је задужена за модел података *User*, тј. прихвата захтеве за упис ових података у базу, и читања из ње.

```
const express = require("express");
const { User } = require("../models/schemas.js");
const router = express.Router();

router.post("/add", async (req, res) => {
  try {
    const newUser = await User.create({
      full_name: req.body.fullName,
      email: req.body.email,
      password: req.body.password,
      admin: req.body.admin,
    });
    await newUser.save();
    return res.send({
      status: 200,
      user: newUser,
    });
  } catch (err) {
    return res.send({
      status: 500,
    });
  }
});

router.get("/find/:email", async (req, res) => {
  try {
    var email = req.params.email;
    const user = await User.findOne({ email: email }).exec();
    return res.send({
      user: user,
      status: 200,
    });
  } catch (err) {
    return res.send({
      status: 500,
    });
  }
});

module.exports = router;
```

Слика 5.1.2.1. Приказ руте *userRoutes* која је задужена за модел података *User*

### 5.1.3. Конфигурација сервера

На серверској страни апликације иницијално су дохватане руте. За сваку од рута су дефинисане и константе које представљају циљну адресу захтева који стижу на сервер. Потом је захваљујући *mongoose* библиотеке урађено повезивање сервера са *MongoDB* базом података.

Након тога је коришћењем програмског оквира *Express* урађена конфигурација и подизање сервера. То је приказано на слици 5.1.3.1. За ову реализацију су употребљене и одговарајуће *middleware* функција, попут већ поменутих *body-parser*-а и *cors*-а. Потом следи постављане наведених рута. На крају је постављен и порт на којем сервер ослушкује захтеве.

```
const express = require("express");
const bodyParser = require("body-parser");
const cors = require("cors");
const corsOptions = {
  origin: "*",
  credentials: true,
  optionSuccessStatus: 200,
};

const app = express();
app.use(cors(corsOptions));
app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());

app.use(USER_SUFFIX, userRoutesHandler);
app.use(COMPANY_SUFFIX, companyRouterHandler);
app.use(USER_COMPANY_SUFFIX, userCompanyRoutesHandler);
app.use(OFFICE_SUFFIX, officeRouterHandler);
app.use(PROJECT_SUFFIX, projectRouterHandler);
app.use(SURVEY_SUFFIX, surveyRoutesHandler);
app.use(RESERVATION_SUFFIX, reservationRoutesHandler);

const PORT = process.env.PORT || 4000;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}.`);
});
```

Слика 5.1.3.1. Конфигурација и подизање сервера

### 5.1.4. Алгоритам расподеле

У овом одељку представљени су изазови на које се наилазило приликом имплементације алгоритма расподеле, као и њихова решења. Прво је приказана реализација задатка да се алгоритам извршава у изабрано предефинисано време. Потом је објашњено и идејно решење самог алгоритма.



i) Извршавање алгорита у одређено време

*Node-cron* је модул који је коришћен да би било омогућено извршавање *Python* скрипте на серверској страни у жељено време. Његова функција *schedule* као аргументе прима задато време и функцију која ће се тада извршити. Време се уноси у виду пет поља: минут, сат, дан у месецу, месец, и дан у недељи, респективно. Уколико се за вредност неког од поља унесе вредност \*, то значи да се алгоритам извршава за све могуће вредности тог поља.

```
var cron = require("node-cron");
callAlgorithm = require("./algorithm/algorithm.js");
cron.schedule("0 20 * * 0-5", async () => {
  | callAlgorithm();
});
```

Слика 5.1.4.1. Извршавање алгорита у 20h

На слици 5.1.4.1. приказан је део кода који омогућава да се позива *Python* скрипта *algorithm.py* сваког радног дана у 20h. За сваку компанију постоји по један *Survey* документ чији је датум за који су се запослени изјаснили једнак наредном радном дану у односу на тренутак извршавања скрипте. Задатак овог фајла је да ишчита све поменуто документе и да за сваки од њих примени алгоритам. Алгоритам ће за сваки *Survey* документ, одн. за сваку компанију генерисати и уписати у базу података *Reservation* документ. Овај документ садржаће низ објеката, где сваки објекат садржи *id* једне канцеларије, и списак запослених којима је додељена та канцеларија.

ii) Идејно решење алгорита

Улазни подаци које прима алгоритам су запослени који су уврстили наредни дан за рад из канцеларије. Они се прихватају у виду низа. Да би алгоритам лакше обрађивао податке, низ поменутих запослених се групише по пројектима на којима раде. Тиме се добија низ низова, где сваки низ представља један пројекат, и садржи инстанце класе *Worker*.

Класа *Worker* се састоји од имејл адресе запосленог, и поља *office*. Она репрезентује запосленог и њему додељену канцеларију. Поље *office* је типа *String* и на почетку је празно. На крају алгорита ће у том пољу бити уписан *id* канцеларије који је алгоритам доделио запосленом.

Подаци који су релевантни за алгоритам су поменути низ низова запослених, и низ канцеларија компаније. Низ канцеларија је низ објеката *Office*. Он представља еквивалент истоимене колекције у бази података. Надаље су приказани кораци које алгоритам извршава:

1. Прво се извршава сортирање *низа низова запослених*, *нерастуће*. У сортираном *низу низова* први ће бити онај низ који има највише запослених. У овом кораку се раду и сортирање канцеларија, од оне која има највише радних места, ка оној која има најмање.
2. Даље се испитују комбинације стављања једног или више низова запослених у канцеларије. Међутим, овде се тражи комбинација која најбоље фитује. То је заправо комбинација таква да после убацивања низа или низова запослених канцеларија буде пуна, тј. да не остане празних места у њој.



Претпоставимо да низова запослених има  $n$ . Тражење свих комбинација би значило разматрање комбинација свих дужина. Комбинација дужине 1 представља испитивање сваког од низа запослених и покушај њиховог стављања у сваку од канцеларија. Комбинација дужине 2 је посматрање свака два низа запослених и разматрање њиховог смештања у сваку од канцеларија, итд. Комбинација дужине  $n$  је заправо анализирање стављања свих низова запослених по пројектима, одн. свих запослени у сваку од канцеларија. У овом процесу тражи се поменута комбинација која најбоље фитује.

Међутим, у овом алгоритму је урађена оптимизација, тако да се не траже комбинације свих дужина. Не разматрају се оне комбинације које немају изгледа да дају добро решење.

Низови запослених по пројектима и низови канцеларија су сортирани нерастуће. Наиме, тражи се највеће  $l$ ,  $l \in [1, n]$  за које важи да је збир првих  $l$  низова запослених мањи од број места у последњој канцеларији. У том случају нема потребе испитивати комбинације дужина од 1 до  $l$ . Ниједна од ових комбинација неће задовољити фитовање неке од канцеларија, јер је број запослених сваке комбинације мањи од најмање канцеларије.

На сличан начин, тражи се најмање  $m$ ,  $m \in [1, n]$  тако да збир последњих  $m$  низова запослених буде већи од броја места у највећој канцеларији. У том случају нема потребе испитивати комбинације дужина од  $m$  до  $n$ . Свака од тих комбинација имаће збир запослених већи од броја места сваке канцеларије, јер је и најкраћа комбинација у старту већа од највеће канцеларије. На овај начин постиже се да се не посматрају комбинације свих дужина  $[1, n]$ . Разматрају се само оне које су дужина  $[l, m]$ , када је то могуће.

У наставку је дата примена ове оптимизације на следећем примеру. Рецимо да се после груписања по пројектима пријављених запослених за одређени дан добија низ низова запослених следећих дужина: 8, 8, 8, 7, 7, 7, 6, 6, 5, 5, 5, 4, 4, 4, 3, 3, 2. С друге стране, нека су величине канцеларија: 35, 33, 27. Тада важи:

- Збир прве три групе запослених је мањи од најмање канцеларије. Стога нема потребе разматрати комбинације дужине 1, 2, 3.
- Збир последњих 10 група запослених је 41. Већи је од капацитета највеће канцеларије. Стога нема потребе гледати комбинације чија је дужина 10 и већа.

Дакле, овај корак алгоритма представља тражење најбољих комбинација. Када се нађе комбинација која би фитовала неку канцеларију, запосленима из те комбинације се додели канцеларија. Ти запослени и канцеларија се више не разматрају. Алгоритам наставља да се рекурзивно извршава све док постоје комбинације које фитују.

3. На овај корак се прелази уколико нема најбоље комбинације, или уколико су примењене све, а остало је још запослених који нису распоређени. Пошто нема више идеалног фитовања, приликом распоређивања више низова запослених по канцеларијама дешаваће се два случаја. Први је да је ће у канцеларији остати неискоришћеног простора. Он може бити узрок другог случаја. Други случај је да један низ запослених мора да се расподели по различитим канцеларијама.

И у овом кораку алгоритма су низови запослених по пројектима и низ канцеларија сортирани нерастуће. Посматра се први низ запослених, тј. онај који има највише запослених на пројекту. Алгоритам тражи сва решења у које све канцеларије би могао да стави тај низ запослених тако да сви стану у једну. За свако решење се прави независна комбинација која се даље рекурзивно извршава. Уколико не могу сви запослени да стану у исту канцеларију, онда се тражи најмањи број канцеларија у које би могли да се распореде сви запослени са пројекта.

Да се не би десило да се превиди најбоље решење, рекурзивно ће се тражити све комбинације. Потом ће оптимално решење бити издвојено коришћењем предефинисане функције цене.

Функција цене се израчунава за сваку комбинацију. Она је иницијално једнака 1. За сваки низ запослених функција цене се множи са одређеним коефицијентом. Уколико су сви запослени са истог пројекта распоређени у исту канцеларију, овај коефицијент биће једнак 1.

Међутим, може да се деси да су запослени са једног пројекта смештени у  $m$  канцеларија. Претпоставимо да тих запослених има  $n$ , и да су распоређени по канцеларијама тако да их је у првој  $n_1$ , у другој  $n_2$ , ... у  $m$ -тој  $n_m$ , где је  $n_1 + n_2 + \dots + n_m = n$ . У овом случају, за сваку од канцеларија где су смештени неки од запослених са посматраног пројекта ће се рачунати предефинисани фактор. Он представља број запослених у тој канцеларији подељен укупним бројем запослених са пројекта. Коефицијент којим ће се помножити функција цене биће једнак производу ових фактора за сваку од поменутих канцеларија. Дакле, коефицијент биће једнак  $\frac{n_1}{n} \cdot \frac{n_2}{n} \cdot \dots \cdot \frac{n_m}{n}$ .

Сваки од прерачунатих фактора облика  $\frac{n_i}{n}$  је мањи од 1. Стога ће и производ свих фактора, одн. тражени коефицијент  $\frac{n_1}{n} \cdot \frac{n_2}{n} \cdot \dots \cdot \frac{n_m}{n}$  бити мањи од 1. Штавише, што је број канцеларија у којима су распоређени запослени са истог пројекта ( $m$ ) већи, то ће коефицијент бити мањи. Како се функција цене множи овим коефицијентима за сваки низ запослених, то ће и она бити мања када се запослени са истог пројекта распоређују у више канцеларија. Дакле, уколико се тражи максимална вредност функције цене, може се закључити да она даје предност оним комбинацијама где су запослени са истог пројекта распоређени у што мањи број канцеларија, а пожељно у једну.

Даље је дат пример употребе функције цене на практичном случају. Узет је пример да је број запослених груписаних по пројектима једнак: 10, 8, 5, а да су величине канцеларија: 14, 11. Ако се не гледају све комбинације, него се само убацују запослени по канцеларија где могу да стану, добило би се решење да се 10 запослених стави у канцеларију величине 14, наредних 8 запослених у канцеларију са 11 места, и последњих 5 би било распоређено у обе. Међутим, посматрањем свих комбинација и избором оне чија је функција цене највећа, алгоритам налази решење да се групе од 8 и 5 запослених ставе у прву канцеларију величине 14, а група од 10 запослених у другу канцеларију величине 11.

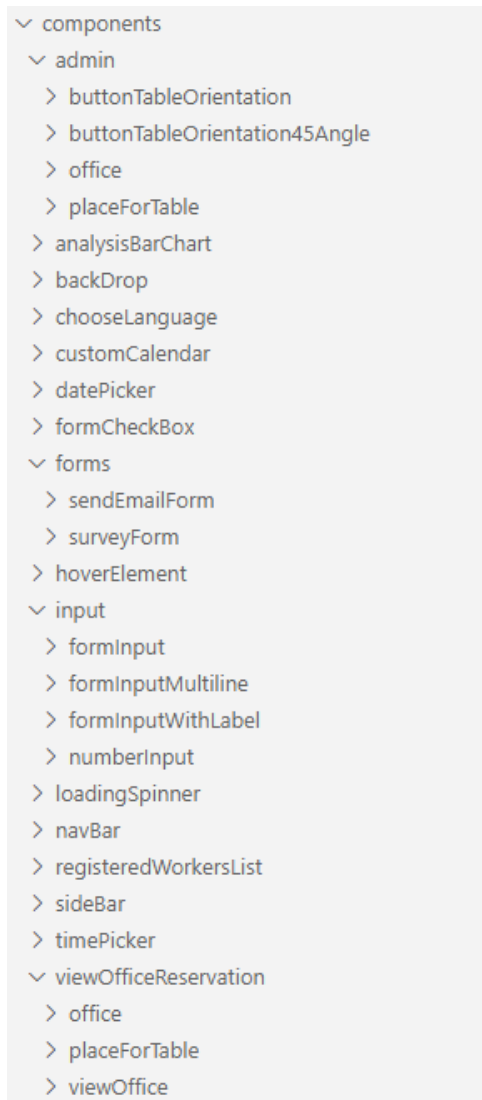
## 5.2. Реализација клијентске стране

У овом одељку приказани су главни изазови и њихова решења приликом имплементације клијентске стране апликације. Надаље су представљене коришћене компоненте и комуникација клијентске и серверске стране.

### 5.2.1. Компоненте

Компонента је основни градивни елементи *React* апликација. Она омогућава да се кориснички интерфејс подели на логичке целине, а потом и да се користи на више места. Компоненте су дефинисане путем два фајла: *.css* и *.jsx*. У првом фајлу се дефинише каква ће бити визуелна репрезентације компоненте на страници. Други фајл представља логику управљања компонентом.

На слици 5.2.1.1. приказан је списак свих реализованих компоненти које су биле потребне за изградњу ове апликације. Поједине компоненте су груписане по директоријумима, ради ефикаснијег организовања кода.



Слика 5.2.1.1. Списак свих компоненти клијентске стране

### 5.2.2. Комуникација клијентске и серверске стране

Сервиси представљају посебне фајлове и класе у којима се реализују различите услужне делатности. У засебним сервисима се имплементира и логика за комуникацију клијентске и серверске стране.

Као што је већ речено, за потребе овог система реализована је база података која има следеће колекције: *User*, *Company*, *UserCompany*, *Office*, *Project*, *Survey*, *Reservation*. Ради интуитивнијег и прегледнијег кода, за све податке који представљају одређену колекцију прављен је засебан сервис за потребну комуникацију са сервером. Сваки од сервиса одговоран је за све неопходне функције које се тичу колекције коју репрезентује.

На слици 5.2.2.1. приказан је један од поменутих имплементираних сервиса - *companyService.js*. Он је задужен за све неопходне обраде над подацима који су еквивалент колекције *Company*.

```
import { COMPANY_SUFFIX, post, put, get, delete_ } from "../baseServerCommunication";

export const addCompany = async (values, setIsLoading, t, successMessage) => {
  | return await post(COMPANY_SUFFIX, values, setIsLoading, t, successMessage);
};

export const updateCompany = async (values, setIsLoading, t, successMessage) => {
  | return await put(COMPANY_SUFFIX, values, setIsLoading, t, successMessage);
};

export const deleteCompany = async (values, setIsLoading, t, successMessage) => {
  | return await delete_(COMPANY_SUFFIX, values, setIsLoading, t, successMessage);
};

export const getCompaniesFromAdmin = async (adminEmail, setIsLoading, t) => {
  | return await get(COMPANY_SUFFIX + "admin/", adminEmail, setIsLoading, t);
};
```

Слика 5.2.2.1. Сервис *companyService.js*

Имплементација *companyService.js* сервиса, као и осталих сервиса за комуникацију са сервером, омогућена је коришћењем помоћних функција фајла *baseServerCommunication.js*. Ту је реализована срж комуникације клијентске и серверске стране. На слици 5.2.2.2. приказан је део тог фајла и имплементација 4 најчешћа типа захтева, а то су: *GET*, *POST*, *PUT*, *DELETE*.

```
const get = async (urlSuffix, param, setIsLoading, t, successMessage = null) => {
  | return await request(GET, urlSuffix, param, null, setIsLoading, t, successMessage);
};

const post = async (urlSuffix, data, setIsLoading, t, successMessage = null) => {
  | return await request(POST, urlSuffix, "", data, setIsLoading, t, successMessage);
};

const put = async (urlSuffix, data, setIsLoading, t, successMessage = null) => {
  | return await request(PUT, urlSuffix, "", data, setIsLoading, t, successMessage);
};

const delete_ = async (urlSuffix, param, setIsLoading, t, successMessage = null) => {
  | return await request(DELETE, urlSuffix, param, null, setIsLoading, t, successMessage);
};
```

Слика 5.2.2.2. Захтеви који се шаљу серверу: *GET*, *POST*, *PUT*, *DELETE*

Поменути захтеви реализовани су захваљујући помоћној предефинисаној функцији *request*. Њена имплементација дата је на слици 5.2.2.3.

Уочено је да се приликом слања захтева *GET*, *POST*, *PUT*, *DELETE* програмски код разликује у другом аргументу функције *fetch*. Стога је функција *request* реализована да се не би исти код писао на више места. Она у зависности од типа захтева шаље одговарајући аргумент *fetchBody*. Такође, у случају успеха или неуспеха захтева приказује адекватну поруку кориснику.

```
const request = async (method, urlSuffix, param, data, setIsLoading, t, successMessage = null) => {
  let result;
  setIsLoading(true);
  let fetchBody;
  if (method === DELETE) {
    fetchBody = {
      method: method,
    };
  } else if (method === GET) {
    fetchBody = {};
  } else {
    fetchBody = {
      method: method,
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify(data),
    };
  }
  await fetch(SERVER_URL + urlSuffix + param, fetchBody)
    .then((response) => {
      return response.json();
    })
    .then((data) => {
      result = data;
      if (result.status === 200) {
        if (!!successMessage) {
          toast.success(successMessage);
        }
      } else {
        toast.error(t("default_server.error." + method));
      }
    })
    .catch((error) => {
      toast.error(t("default_server.error." + method));
    });
  setIsLoading(false);
  return result;
};
```

Слика 5.2.2.3. Помоћна функција *request*

## 6. ЗАКЉУЧАК

Пандемија изазвана вирусом Ковид-19 у потпуности је променила свет који смо некад познавали. С променама епидемиолошке ситуације мењале су се и начини пословања свих делатности, посебно оних канцелријског типа. Поред традиционалног модела рада који подразумева одлазак у канцеларију, у пракси је виђен и удаљен рад. Примењиван је и хибридни модел рада који представља комбинацију претходна два када је здравствена ситуација то дозвољавала.

С обзиром на то да је прошло више од две и по године од почетка пандемије, на површину су испливале све предности и мане сваког од модела рада. Испоставило се да највећем броју запослених управо одговара хибридни модел. Штавише, запосленима је један од критеријума за избор компаније где ће радити постала управо могућност избора да раде из канцеларије или удаљено у зависности како им одговара.

Последица хибридног начина рада је губитак потреба да сваки запослени има своје радно место. Наиме, неће дешавати да сви запослени буду истовремено присутни. Уместо тога, уз резервацију и дељење места може се постићи да сви запослени који су изразили жељу за рад из фирме за одређени дан имају своје место. Штавише, тако су и компаније на добитку јер им се отвара велика уштеда у простору и ресурсима. Сходно томе, циљ ове тезе је био реализација система који ће дати одговор на изазове хибридног рада и аутоматизовати управљање радним простором.

Попут разних апликација сличне намене, и у реализацији овог система је подржана функционалност изјашњавања запослених када желе да раде из канцеларије. Она је реализована у виду календара. Кликком на жељени датум, запослени може да се пријављује за долазак у канцеларију, одн. да откаже долазак. Такође, за сваки селектовани датум се излистава и списак свих запослених који су се до тог тренутка пријавили за долазак у канцеларију за тај дан. Штавише, постоји и могућност да се за изабраног колегу прегледа његов календар, те запослени може да ажурира своје доласке са другима.

Што се администраторске стране тиче, реализоване су функционалности попут позивања запослених у апликацију путем имејла, доделе пројеката и задужења, креирање канцеларија и њихова организација по спратовима компаније. Постоји и приказ груписања запослених по канцеларијама за изабрани датум. Имплементирана је и јако важна функционалност за аналитику попуњености канцеларије у претходном периоду. На овај начин управа компаније може одлучивати о евентуалном повећању или смањењу простора у зависности од навика својих запослених.

Апликација ради по принципу да су пријаве за долазак у канцеларију за сутрашњи дан отворене до неког предефинисаног времена. Након тога се извршава алгоритам који распоређује запослене по канцеларијама, тако да људи из истог тима буду у истој канцеларији ако је то физички могуће. На овај начин динамички се прави најбољи распоред за сваки дан на основу података о томе ко ће бити присутан.

Од техничке подршке изабрано је да клијентска страна буде реализована уз помоћ *React* библиотеке. За имплементацију серверског дела апликације одабрано је *Node.js*

окружење са програмским оквиром *Express*. За трајно чување података изабрана је *MongoDB* база. Што се алгоритма расподеле запослених тиче, он је реализован као *Python* скрипта која се покреће у 20h и генерише распоред седења за наредни дан.

Мана оваквих система је то што стално морају да прате корак с временом. Уколико не испуне жеље корисника и компанија брзо застаревају. Стога се и за пројекат описан овим документом разматра увођење додатних функционалности.

Једна од ставки која би свакако требало да буде подржана у каснијим верзијама апликације је позивање радника на састанке, и резервисање просторије за њихово одржавање. Између осталог, било би од значаја да постоји и евиденција попуњености радних места у реалном времену. Тако би се постигла још већа искоришћеност простора. Штавише, запослени који нису попунили анкету за долазак за одређени дан могли би накнадно да резервишу своје место непосредно пред долазак.

Да би се постигла највиша продуктивност запослених, од значаја је да и њихово време за паузу и рекреацију буде оптимално организовано. Стога су функционалности које би биле од значаја свакако и резервисање просторија за рекреацију, организовање *team building*-а, као и могућност наручивања хране и пића.

Системи попут овог немају крајњи циљ, већ потребу за сталним проширењем како би се корисницима пружиле што боље услуге. У апликацији описаној овим радом имплементиране су функције које представљају срж једног система за управљање радним простором. Ипак, цео пројекат је прављен модуларно, са идејом да напредује и постане конкурентан на тржишту.

## ЛИТЕРАТУРА

- [1] V. Subramanian, *Pro MERN Stack*, Apress, 2017.
- [2] F. Zammetti, *Practical React Native*, Apress, 2018.
- [3] F. Doglio, N. Sabharwal, *REST API Development with Node.js*, Apress, 2018.
- [4] S. G. Edward, N. Sabharwal, *Practical MongoDB*, Apress, 2015.
- [5] React Documentation [Online]. Available at: <https://reactjs.org/> (05.09.2022.)
- [6] Node API Reference Documentation [Online]. Available: <https://nodejs.org/en/docs/> (05.09.2022.)
- [7] Express Documentation [Online]. Available: <http://expressjs.com/en/4x/api.html> (05.09.2022.)
- [8] Node Cron Documentation [Online]. Available: <https://www.npmjs.com/package/node-cron> (05.09.2022.)
- [9] MongoDB Documentation [Online]. Available: <https://www.mongodb.com/docs/> (05.09.2022.)
- [10] Mongoose API Documentation [Online]. Available: <https://mongoosejs.com/docs/api.html>
- [11] PyMongo Documentation [Online]. Available: <https://www.mongodb.com/docs/drivers/pymongo/>
- [12] Firebase Documentation [Online]. Available: <https://firebase.google.com/docs> (05.09.2022.)
- [13] EmailJS Documentation [Online]. Available: <https://www.emailjs.com/docs/> (05.09.2022.)



## СПИСАК СКРАЋЕНИЦА

SPA	<i>Single Page Application</i>
JSON	<i>JavaScript Object Notation</i>
CORS	<i>Cross-Origin Resource Sharing</i>
HTTP	<i>Hypertext Transfer Protocol</i>
REST	<i>REpresentational State Transfer</i>
BSON	<i>Binary JavaScript Object Notation</i>

## СПИСАК СЛИКА

Слика 4.1.1.1. Главни кориснички екран администратора .....	14
Слика 4.1.2.1. Приказ свих канцеларија .....	15
Слика 4.1.2.2. Креирање канцеларије.....	15
Слика 4.1.3.1. Страница са детаљима о пројектима .....	16
Слика 4.1.4.1. Списак запослених .....	16
Слика 4.1.4.2. Корисници позвани у апликацију .....	17
Слика 4.1.4.3. Позивање запосленог путем имејла у апликацију.....	17
Слика 4.1.5.1. Избор датума.....	18
Слика 4.1.5.2. Избор канцеларије и приказ распореда седења за одабрану канцеларију .....	18
Слика 4.1.5.3. Списак запослених у изабраној канцеларији за изабрани датум.....	19
Слика 4.1.5.4. Тлоцрт канцеларије .....	19
Слика 4.1.6.1. Анализа попуњености .....	20
Слика 4.2.1.1. Страница са детаљима о пројектима и колегама.....	21
Слика 4.2.2.1. Резервација радног места.....	22
Слика 4.2.2.2. Претрага распореда долазака од колега .....	23
Слика 4.2.3.1. Страница са приказом распореда седења по канцеларијама.....	24
Слика 5.1.1.1. Модели података .....	26
Слика 5.1.2.1. Приказ руте <i>userRoutes</i> која је задужена за модел података <i>User</i> .....	27
Слика 5.1.3.1. Конфигурација и подизање сервера .....	28
Слика 5.1.4.1. Извршавање алгоритма у 20h .....	29
Слика 5.2.1.1. Списак свих компоненти клијентске стране.....	32
Слика 5.2.2.1. Сервис <i>companyService.js</i> .....	33
Слика 5.2.2.2. Захтеви који се шаљу серверу: <i>GET, POST, PUT, DELETE</i> .....	33
Слика 5.2.2.3. Помоћна функција <i>request</i> .....	34