

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН Рафетов 18510



НАЦИОНАЛНА ПРОФЕСИОНАЛНА ГИМНАЗИЯ ПО
КОМПУТЕРНИ ТЕХНОЛОГИИ И СИСТЕМИ – гр. ПРАВЕЦ ПРИ
ТУ – СОФИЯ

ДИПЛОМНА РАБОТА

ТЕМА : *РАЗРАБОТВАНЕ НА ПЧЕЛАРСКА УЕБ ПЛАТФОРМА*

ДИПЛОМАНТ :

Джан Рафетов

РЪКОВОДИТЕЛ:

доц. д-р инж. Явор Томов

КУРСОВ НОМЕР : *18510*

СПЕЦИАЛНОСТ : *код 4810201 “Системно програмиране“*

ПРАВЕЦ

2023

Съдържание

| | |
|---|-----------|
| Глава 1 - УВОД | 3 |
| Глава 2 - ИЗПОЛЗВАНИ ТЕХНОЛОГИИ | 4 |
| 2.1.1 IntelliJ IDEA | 4 |
| 2.1.2 Java Spring Boot | 5 |
| 2.1.3 REST API | 7 |
| 2.1.4 Postman API | 7 |
| 2.1.5 Hibernate и JPA | 8 |
| 2.1.6 MySQL | 9 |
| 2.1.7 HTML | 9 |
| 2.1.8 CSS | 10 |
| 2.1.9 Bootstrap | 11 |
| 2.1.10 JavaScript | 12 |
| 2.1.11 Thymeleaf | 12 |
| Глава 3 - ИЗПОЛЗВАНИ АРХИТЕКТУРИ | 14 |
| 3.1 Запознаване с Модулите | 14 |
| 3.2 Структура на приложението | 15 |
| 3.2.1 Обикновен Потребител | 15 |
| 3.2.2 Администраторски Потребител | 16 |
| 3.2.3 Използвани dependency-та | 18 |
| Глава 4 - РАЗРАБОТВАНЕ НА ПЧЕЛАРСКА УЕБ ПЛАТФОРМА | 19 |
| Глава 5 - ПРИЛОЖЕНИЕ НА ПЧЕЛАРСКА УЕБ ПЛАТФОРМА | 34 |
| 5.1 Функционалност за регистриране на потребител | 34 |
| Глава 6 - ИЗВОДИ И ЗАКЛЮЧЕНИЕ | 46 |
| Глава 7 - ИЗПОЛЗВАНА ЛИТЕРАТУРА | 47 |
| Глава 8 - СПИСЪК С ИЗПОЛЗВАННИТЕ ОЗНАЧЕНИЯ И СЪКРАЩЕНИЯ. | 48 |

Глава 1 - УВОД

Настоящата дипломна работа има за цел разработка на уеб платформа, която служи за продажба на пчеларски артикули(пчелен мед, восък, отводки , кошери и други).

Причината темата на курсовата работа да е пчеларство ,е че в България има голям набор хора, които се занимават с тази работа и те постоянно си търсят артикули(пчеларски инвентар, техника и други) от интернет.

Има много голяма общност в социалните мрежи и форуми, които се опитват от там да си закупят или продадат пчеларските продукти, но понеже тези социални мрежи и форуми не са предназначени за продажба на пчелни продукти се изпитват много затруднения, понеже не са електронен магазин за пчеларски стоки.

Задачата е да се напише софтуер с най-актуалните и подходящи технологии за изработката на електронен магазин в конкретна сфера(пчеларство).

Очакванията на нашата платформа е да достигне до пчелари, които искат да си купят пчеларски стоки от интернет.

Уеб приложението ще има основните функции за такъв тип приложение, като потребителят ще може да си създаде профил да качва и разглежда обяви.

Глава 2 - ИЗПОЛЗВАНИ ТЕХНОЛОГИИ

В настоящият проект са използвани следните технологии:

Spring Boot, Hibernate, JPA, Postman, MySQL, HTML, CSS, Bootstrap, JavaScript, Thymeleaf.

За среда за разработка е използвана: IntelliJ IDEA 2021.1(Ultimate Edition),

За генериране на структура на Spring Boot проекта е използван: Spring Initializr

За операционна система : Mac OS Monterey 12.5,

За Хостинг се използва: локален хост(с порт 8080)

Нека да разгледаме използваните технологии по-подробно:

2.1.1 IntelliJ IDEA

IntelliJ IDEA 2021.1(Ultimate Edition) е интегрирана среда за разработка (IDE) за езици JVM, предназначена да увеличи производителността на разработчиците. Тя изпълнява рутинните и повтарящи се задачи вместо вас, като осигурява интелигентно попълване на код, статичен анализ на кода и рефакторинг, и ви позволява да се съсредоточите върху светлата страна на разработването на софтуер, като го прави не само продуктивно, но и приятно преживяване.

Има още много плюсове като:

IntelliJ IDEA Ultimate може да ви помогне да разработват цялостни уеб приложения благодарение на мощните си интегрирани инструменти, поддръжката на JavaScript и свързаните с него технологии, както и разширената поддръжка на популярни фреймворки като Spring, Spring Boot, Jakarta EE, Micronaut, Quarkus, Helidon. Освен това можете да разширите IntelliJ IDEA с безплатни плъгини, разработени от JetBrains, които ви позволяват да работите с други езици за програмиране, включително Go, Python, SQL, Ruby и PHP.

Предлага много голяма ергономичност на разработчика:

Той осигурява бърз достъп до всички функции и интегрирани инструменти, които са важни за работата ви, както и широк набор от възможности за персонализиране. Можете да настроите всичко така, че да подпомага вашия работен процес: да зададете

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

преки пътища, да инсталирате приставки, да персонализирате интерфейса по свой вкус и др.

Също така Преструктурирането на съществуващия код, без да се променя неговата функционалност, рефакторинг. IntelliJ има по-добра поддръжка за рефакториране от другите среди за разработка. IntelliJ предоставя няколко функции за рефакториране на кода, като например преименуване, преместване и изтриване.

2.1.2 Java Spring Boot

Spring Boot по същество е разширение на рамката Spring, което елиминира шаблонните конфигурации, необходими за създаването на Spring приложение. Тя позволява писането на микроуслуги на езика Java. Spring Boot идва с Tomcat като уеб сървър и много други допълнителни функции.

С помощта на Spring Boot можете да разработвате по-бързо и с по-малко загуба на време, като пишете по-малко код за бази данни. Той предлага използване на REST(Representational State Transfer) за крайни потребители на услуги и разполага с мощна REST конфигурация. Много от конфигурациите се извършват автоматично.

Java Spring

Казано по-просто, рамката Spring осигурява цялостна инфраструктурна поддръжка за разработване на Java приложения. Той е снабден с някои хубави функции като Dependency Injection и готови модули като: Spring JDBC ,Spring MVC ,Spring Security ,Spring AOP ,Spring ORM ,Spring Test

| РАЗЛИКИ | Spring | Spring Boot |
|-------------------|---|---|
| Къде се използва? | използва за изграждане на приложения. | главно за разработване на REST API. |
| Конфигурации | трябва да направите конфигурациите ръчно. | Съществуват конфигурации по подразбиране, които позволяват по-бързо стартиране. |

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”

ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

| | | |
|---------------------------|---|---|
| Boilerplate(шаблонен) код | изиска твърде много редове код (boilerplate code) дори за минимални задачи. | Избягвате шаблонния код, което намалява времето и увеличава производителността. |
| Плъгини | не предоставя плъгини за maven, Gradle и др. | предоставя плъгини за инструменти за изграждане за Maven и Gradle. |

Spring Boot, Spring

Spring Boot не е предназначен да замени Spring, по-скоро има за цел да направи работата с него по-бърза и по-лесна. В резултат на това по-голямата част от модификациите, необходими за миграцията на дадено приложение, са свързани с конфигурацията.

Микроуслуги в Spring Boot

Те представляват работа на всяка операция на услугата в отделен процес. По този начин това е архитектура за разработване на софтуер, която разпределя натоварването върху основния процес на изпълняваната програма и позволява тя да се изпълнява в отделни процеси, като по този начин позволява разработването на по-производителни услуги.

Предимства на Spring Boot

- о Създава самостоятелни Spring приложения, които могат да се стартират с помощта на Java
- о Предоставя готови за работа функции като метрики, проверки на състоянието и външна конфигурация.
- о Няма изискване за XML конфигурация.
- о Увеличава производителността и намалява времето за разработка.

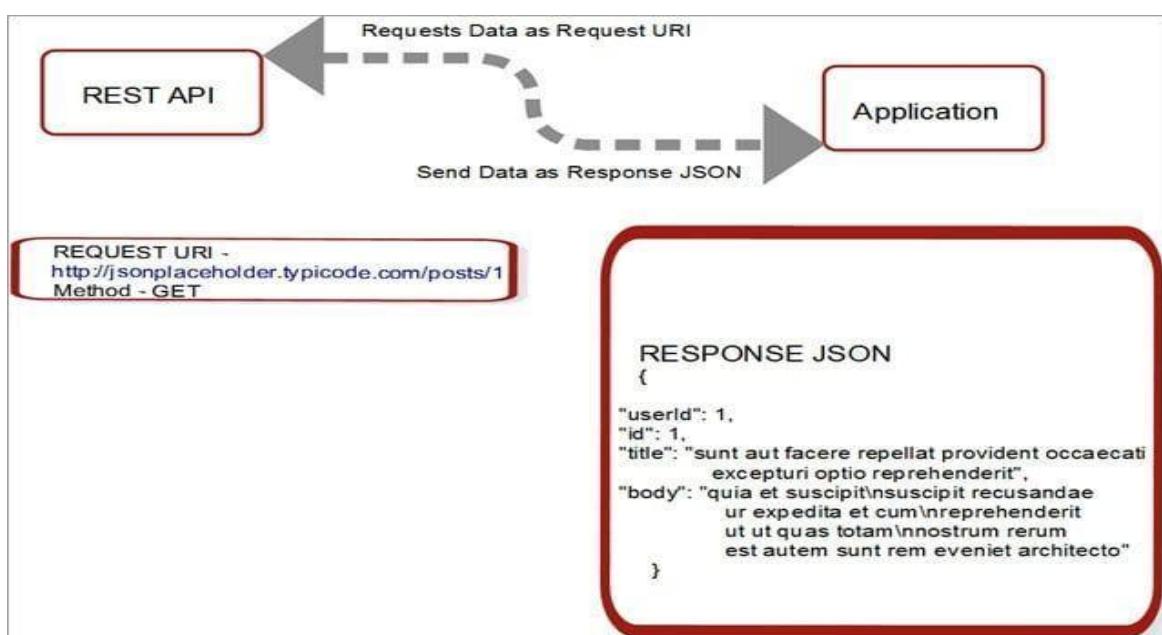
Ограничения на Spring Boot

- о Spring Boot може да използва dependencies, които няма да се използват в приложението. Тези dependencies увеличават размера на приложението.

REST покрива принципите на уеб, включително неговата архитектура, предимства и всичко останало. REST бързо се превърна във фактически стандарт за изграждане на уеб услуги в мрежата, тъй като те са лесни за изграждане и лесни за използване. Какви са ползите? Мрежата и нейният основен протокол - HTTP - предоставят множество функции: Подходящи действия (GET, POST, PUT, DELETE, ...), Кеширане, Пренасочване и препращане, Сигурност (криптиране и удостоверяване)

2.1.3 REST API

Rest API е интерфейс, който се използва от софтуерните компоненти за комуникация помежду си. Услугата е функция, която е добре дефинирана, самодостатъчна и не зависи от други услуги. Уеб услугата е вид API, почти всички от тях работят през HTTP. Когато се разработи уеб API с помощта на REST Architecture, тогава той се нарича REST Web API. Примерна REST Архитектура:



Фигура 2-1 Снимка на примерна REST Архитектура

2.1.4 Postman API

Postman е самостоятелна платформа за тестване на софтуерни интерфейси API (Application Programming Interface) за изграждане, тестване, проектиране, модифициране и документиране на API. Тя представлява прост графичен потребителски интерфейс за изпращане и разглеждане на HTTP заявки и отговори.

При използване на Postman за целите на тестването не е необходимо да се пише мрежов код на HTTP клиент. Вместо това изграждаме набори от тестове, наречени колекции, и позволяваме на Postman да взаимодейства с API.

В този инструмент са вградени почти всички функционалности, от които може да се нуждае всеки разработчик. Този инструмент има възможност да извършва различни видове HTTP заявки, като GET, POST, PUT, PATCH

Postman API

- Използване на колекции - Postman позволява на потребителите да създават колекции за своите API-извиквания. Всяка колекция може да създава множество заявки и под папки. Това ще ви помогне да организирате тестовите пакети.
- Разработване на тестове - За да се тестват контролните точки, към всяко API повикване трябва да се добави проверка на състоянието на успешен HTTP отговор.
- Отстраняване на грешки - За ефективно отстраняване на грешки в тестовете конзолата на Postman помага да се проследи какви данни се извличат.
- Непрекъсната интеграция - може да поддържа непрекъсната интеграция.

2.1.5 Hibernate и JPA

Hibernate

Hibernate е Java рамка и ORM (Object Relational Mapping) инструмент, който се използва за реализиране на методите на JPA.

JPA

JPA(Java Persistence API) е като мост между моделите на приложението Spring и релационната база данни, който се използва за управление и достъп до данните между обектно ориентираните модели на приложението Spring и базата данни.

Казано по-просто, ако не използваме JPA, трябва да дефинираме всички методи за вмъкване на записи в таблицата на MySQL. JPA не извършва никакви специфични операции, а предоставя различни видове методи без имплементация.

Начин на работа на JPA:

JPA е абстракция, която се използва за съпоставяне на java обект с базата данни. Тя съдържа различни структури на методите, които се използват за манипулиране на записите в таблицата, и също така предоставя SQL заявки за конкретни операции.

2.1.6 MySQL

ОПРЕДЕЛЕНИЕ

MySQL е система за управление на релационни бази данни разработена от Oracle, която се основава на структуриран език за заявки (SQL). Базата данни е структурирана колекция от данни. Може да бъде всичко - от обикновен списък за пазаруване до галерия с картини или място за съхраняване на огромните количества информация в корпоративна мрежа. По-конкретно, релационната база данни е цифрово хранилище, събиращо данни и организиращо ги според релационния модел. В този модел таблиците се състоят от редове и колони, а връзките между елементите на данни следват строга логическа структура

2.1.7 HTML

ОПРЕДЕЛЕНИЕ

Това е абревиатура от Hyper Text Markup Language. Използва за описание на структурата на даден документ използвайки маркери – така наречените тагове. Един HTML документ е направен от много HTML тагове и всеки HTML таг съдържа различно съдържание.

HTML Начин на работа

Когато някой кликне на линк в страница, или въведе нов URL в адресното поле, браузърът изпраща заявка за документ от Web server-а, където е качена страницата, посредством HyperText Transport Protocol (HTTP). Сървърът след това изпраща документа обратно до потребителя като се визуализира на дисплея от браузъра. Всички

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

неша, които са на този документ – текст, снимки, аудио или видео файлове, всички те са подредени посредством HTML структура.

HTML елемент

HTML елемент се определя от начален таг, малко съдържание и краен таг:

<име на тага> съдържание </ име на тага >

И ето някои важни елементи на HTML

| | |
|------------|--|
| <!DOCTYPE> | Определя типа документ или инструктира браузъра относно версията на HTML. |
| <html> | Този таг информира браузъра, че това е HTML документ. |
| <head> | Трябва да е първият елемент в элемента <html>, който съдържа метаданните (информация за документа). Трябва да се затвори, преди етикетът на тялото да се отвори. |
| <body> | Този таг съдържа основното съдържание на HTML документа. |
| <h1> | Текстът между тага <h1> описва заглавието от първо ниво на уеб страницата. |

Структура на една HTML страница

```
<html>
  <head>
    <title>Page title</title>
  </head>

  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
  </body>
</html>
```

Фигура 2-2 Снимка на примерна структура на HTML страница

2.1.8 CSS

ОПРЕДЕЛЕНИЕ

CSS Cascading Style Sheets е език за описание на стилове. Преди създаването на езика, задаване на стилове с HTML тагове е било кошмарът на разработчиците. Това е било невероятно предизвикателство при изграждането на големи сайтове. Именно това е причина да се разработи CSS – стиловете излизат от обхвата на HTML, улесняват процеса по работа и реално това позволява всички стилове да бъдат променени бързо и с лекота. Чрез CSS се отделя описанието на съдържанието и структурата на сайтовете от стилизирането им.

Тя променя начина на внедряване на някои визуални елементи и изобразяването им.

CSS се състои от правила за стил, които се интерпретират от браузъра и след това се прилагат към съответните елементи в документа. Правилото за стил се състои от три части.

- Селектор - Селекторът е HTML таг, към който ще се прилага даден стил.
- Свойство - Свойството е вид атрибут на HTML таг. Най-просто казано, всички HTML атрибути се превръщат в CSS свойства. Те могат да бъдат цвят, рамка и т.н.
- Стойност - Стойностите се присвояват на свойствата. Например свойството color може да има стойност red или #F1F1F1 и т.н

2.1.9 Bootstrap

ОПРЕДЕЛЕНИЕ

Bootstrap е сред най-популярните HTML, CSS и JS рамки. Това е безплатен набор от инструменти за създаване на уеб сайтове, който включва типографски елементи, готови уеб форми, бутони, менюта и други уеб елементи. Основният фокус на рамката признава развитието на сложни мобилни проекти. Важно предимство на Bootstrap е голяма общност, която е в пъти по-голяма от тази на конкурентите си. Основната характеристика на Bootstrap е, че тя не е само CSS рамка, но също така и Javascript библиотека. Разработени са готови за използване стилове и скриптове, които са свързани с документа чрез писане на необходимите класове и атрибути на HTML

елементи. Платформата, помага да се развие структурата на софтуерна система, която улеснява разработването и интегрирането на отделни компоненти на голям проект. Ускорява и улеснява процеса на работа.

2.1.10 JavaScript

ОПРЕДЕЛЕНИЕ

JavaScript е скриптов език за създаване на динамично съдържание на уеб страници. Той създава елементи за подобряване на взаимодействието на посетителите на сайта с уеб страниците, като падащи менюта, анимирани графики и динамични фонови цветове. Разработчиците обикновено използват JavaScript заедно с HTML и CSS.

Предимства на JavaScript

- + Опростеност - простата структура улеснява изучаването и прилагането на JavaScript
- + Бързина - JavaScript изпълнява скриптове директно в уеб браузъра, без да се свързва първо със сървър или да се нуждае от компилатор.
- + Универсалност - JavaScript е съвместим с други езици като PHP, Perl и Java.
- + Актуализации - екипът за разработване на JavaScript и ECMA International непрекъснато актуализират и създават нови рамки и библиотеки, което гарантира тяхната актуалност в индустрията.

Недостатъци на JavaScript

- Съвместимост с браузъри - различните уеб браузъри интерпретират кода на JavaScript по различен начин, което води до непоследователност.
- Сигурност - кодът на JavaScript, който се изпълнява от страната на клиента, е уязвим за използване от безотговорни потребители.
- Отстраняване на грешки - макар че някои редактори на HTML поддържат отстраняване на грешки, те са по-малко ефективни от други редактори. Тъй като браузърите не показват никакви предупреждения за грешки, намирането на проблема може да се окаже предизвикателство.

2.1.11 Thymeleaf

ОПРЕДЕЛЕНИЕ

Thymeleaf е Java шаблонен език и библиотека за трансформация на шаблони. Той е проектиран да бъде модерен, безопасен и изключително ефективен за използване в уеб приложения. Thymeleaf използва HTML5 за свои шаблони и може да работи както на страната на сървъра, така и на клиентската страна.

Той предоставя много различни функционалности, включително изрази, условни изрази, цикли и много други. Thymeleaf е лесен за използване и позволява бързо и лесно интегриране във вашия Java проект. Той се използва широко в уеб разработката, особено в Spring приложения, като е съвместим с множество инструменти и технологии, което го прави много гъвкав за използване.

Предимства на Thymeleaf

- + Лесен за използване: Thymeleaf е лесен за използване и има прост синтаксис, който е подобен на HTML.
- + Модерен и безопасен: Thymeleaf използва съвременни технологии и е безопасен по отношение на XSS атаки, тъй като поддържа HTML екраниране по подразбиране.
- + Гъвкавост: Thymeleaf е гъвкав и може да се използва в различни сценарии на уеб разработката. Той поддържа различни шаблонни режими и може да се използва както на страната на клиента, така и на страната на сървъра.
- + Интеграция с Java приложения: Thymeleaf е направен да работи с Java приложения и има силна интеграция със Spring Framework.
- + Кеширане на шаблони: Thymeleaf може да кешира шаблоните, за да ускори зареждането на уеб страници.

Недостатъци на Thymeleaf

- Скорост: Thymeleaf може да бъде по-бавен в сравнение с други шаблонни езици като JSP или Velocity, тъй като използва динамично генериране на HTML.
- Сложност: Thymeleaf може да бъде малко по-сложен от други шаблонни езици, тъй като има повече опции и настройки.
- Изиска умения в HTML: Thymeleaf изиска знания в HTML, за да се създават шаблони. За хора без опит в уеб разработката, може да бъде трудно да се научат.
- Интеграция с други технологии: Thymeleaf може да бъде трудно да се интегрира с някои други технологии, особено ако не са направени да работят заедно.
- Сложност на изпълнение на някои функции: Някои функции на Thymeleaf могат да бъдат по-сложни за изпълнение в сравнение с други шаблонни езици, тъй като изискват по-голяма конфигурация и подготовка.

Глава 3 - ИЗПОЛЗВАНИ АРХИТЕКТУРИ

3.1 Запознаване с Модулите

Проектът е със средно ниво на сложност, но над знанията изискващи се за ученик. Изискват се познания по HTML, CSS, Bootstrap, JavaScript, Java, Spring, MySQL и други. Проектът е логически свързан и структуриран, за което се изиска комбиниране на различни технологии и познания в спецификата на всички използвани технологии.

В приложението има няколко модула. Първият от тях е този, който е видим за обикновения потребител. Когато се въведе информация в дадено поле (примерно при регистрация) първият модул взема тази информация и я предава на следващия модул, но първият модул може и да извежда информация, която вече е налична.

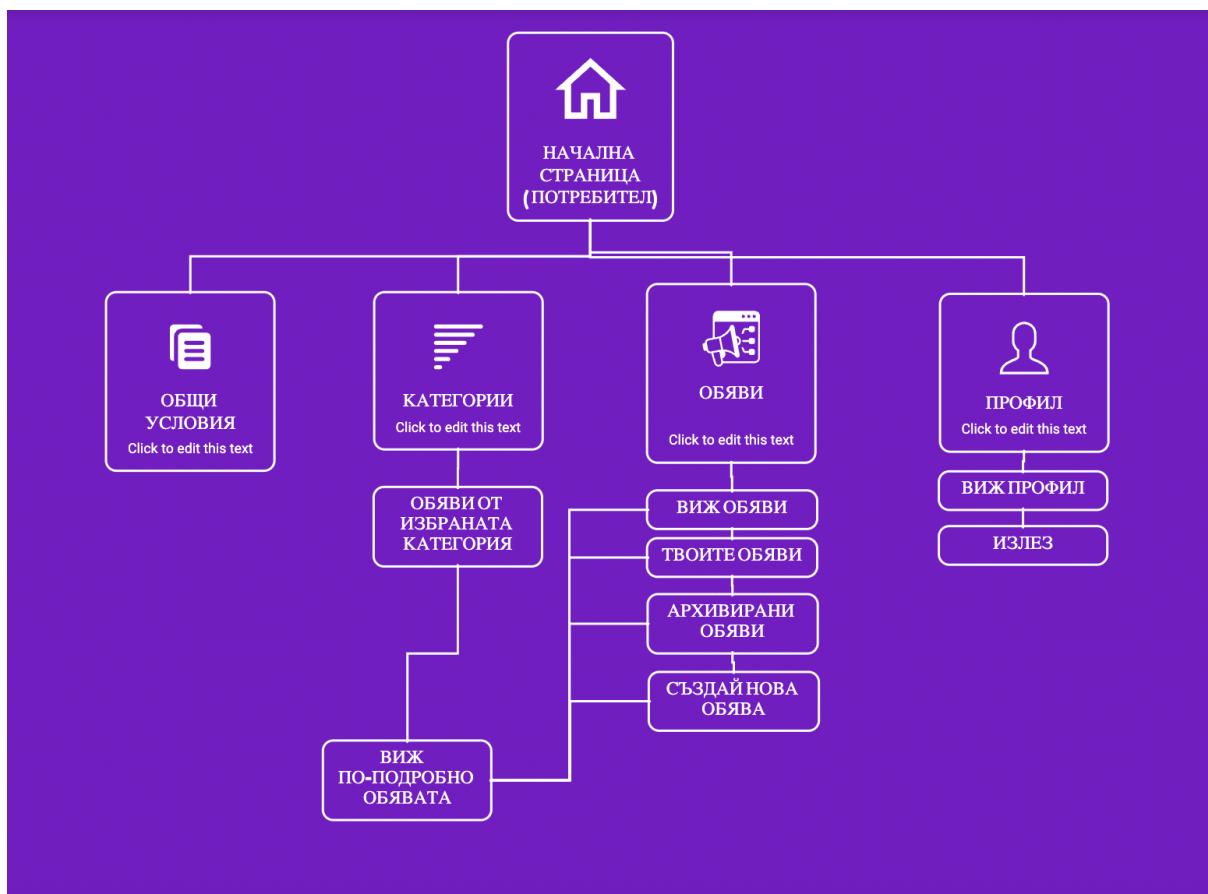
Вторият модул взима информацията от първия и прави основни валидации върху информацията, която е получена (например всички полета ли са попълнени, брой символи в името и т.н), ако всичко е наред – информацията се изпраща на следващия слой. Следващият слой обработва тази информация и я преобразува в такава, че базата с данни да я запамети и вече да се създаде потребител.



Фигура 3-1 Снимка на модулите

3.2 Структура на приложението

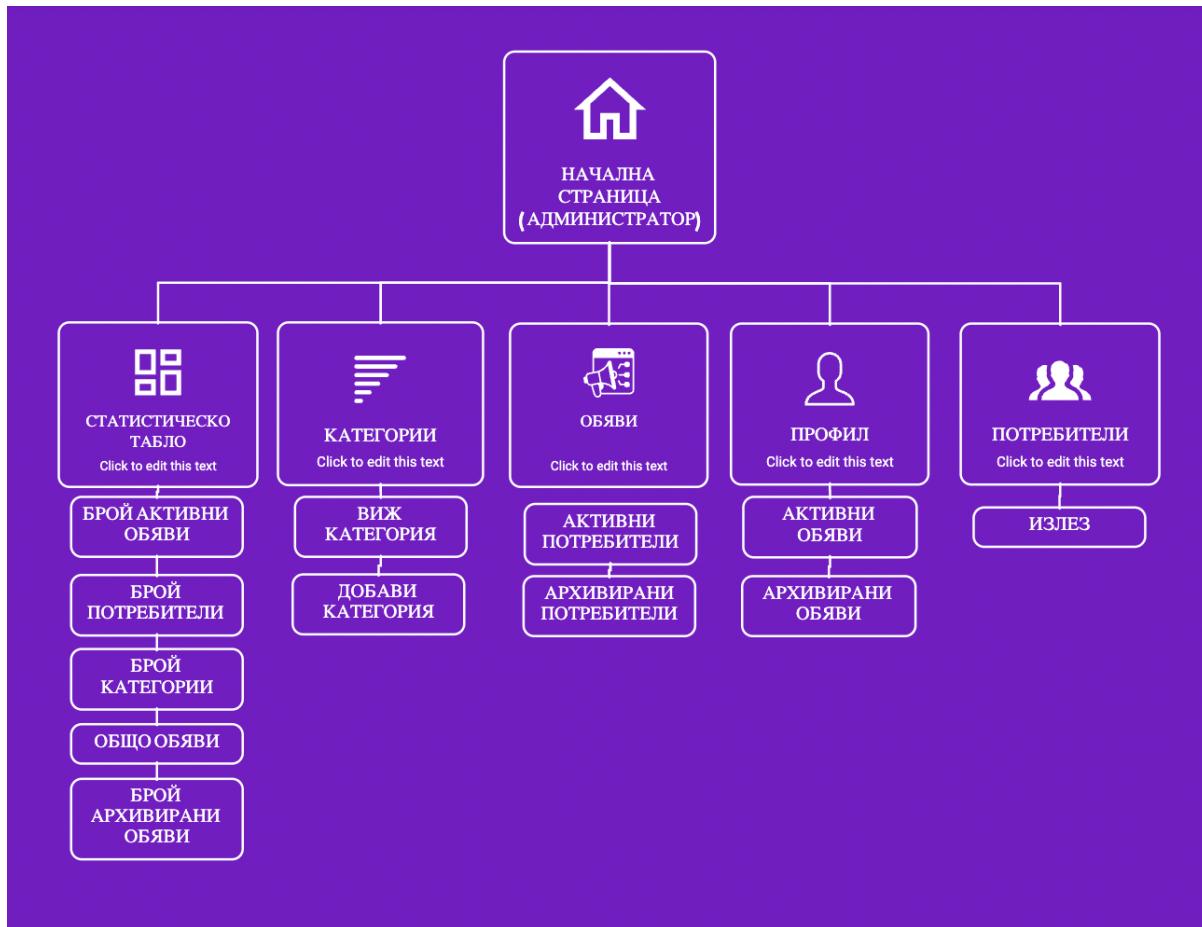
3.2.1 Обикновен Потребител



Фигура 3-2 Карта на сайта за обикновен потребител

1. Може да види всички категории и обявите, съответстващи на категорията.
2. Може да види всички активни обявии.
3. Може да види своите активни обявии и да ги архивира.
4. Може да види своите архивирани обявии.
5. Може да създаде обява

3.2.2 Администраторски Потребител



Фигура 3-3 Карта на сайта за администратор

1. Може да види статистиката на сайта(брой активни обяви ,потребители ,категории и др.).
2. Може да види всички категории и да ги изтрива.
3. Може да добави категория
4. Може да види всички активни потребители и да ги деактивира
5. Може да види всички деактивирани потребители и да ги активира или да изтрие
6. Може да види всички активни обяви и да ги деактивира
7. Може да види всички архивирани обяви и да ги активира или да ги изтрие

3.2.3 Използвани dependency-та

(Намират се в “pom.xml” файла)

,“org.springframework.boot:spring-boot-starter-data-jpa“

,“org.springframework.boot:spring-boot-starter-email“

,“org.springframework.boot:spring-boot-starter-security“

,“org.springframework.boot:spring-boot-starter-thymeleaf“

,“org.springframework.boot:spring-boot-starter-validation“

,“org.springframework.boot:spring-boot-starter-web“

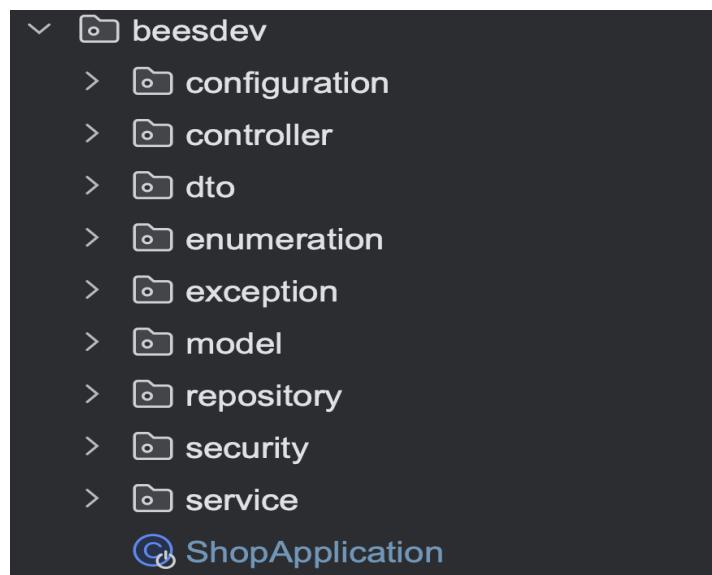
,“org.springframework.boot:spring-boot-devtools“

,“org.springframework.boot:mysql-connector-java“

Глава 4 - РАЗРАБОТВАНЕ НА ПЧЕЛАРСКА УЕБ ПЛАТФОРМА

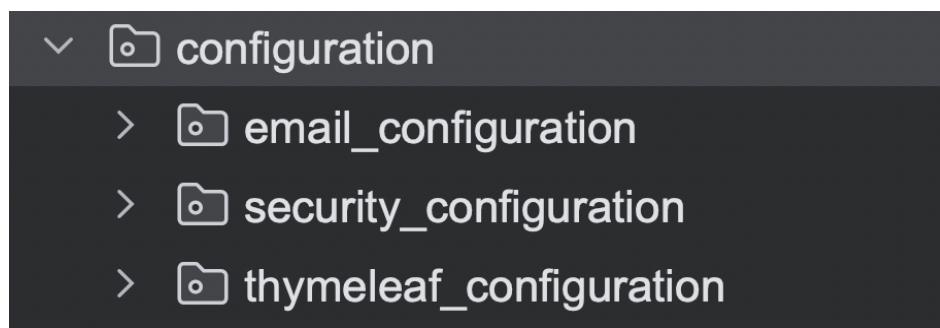
Основната технология използваната в приложението е Java. Структурата е стандартна структура на Spring Framework (MVC MODEL VIEW CONTROLLER).

Тук е частта на бекенда на приложението



Фигура 4-1 Снимка на файловата структура на бек-енда

В папката „configuration“ се съдържа цялата конфигурация на приложението.



Фигура 4-2 Снимка на файловата структура на конфигурационни файлове на бек-енда

В папката „email_configuration“ е конфигурацията ни за имейл сървъра, за изпращане на имейли:

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

Тук виждаме конфигурационен клас, който служи за изпращане на имейли:

@Configuration анотацията указва, че класът е конфигурационен клас, който ще се използва за конфигуриране на приложението.

@Value анотациите се използват за вземане на стойностите на настройки от файлове за конфигурация, които се използват за настройка на JavaMailSenderImpl.

@Bean анотацията указва, че методът mailSender() връща bean, който може да се използва в приложението.

JavaMailSender е интерфейс, който дефинира методи за изпращане на електронни съобщения. JavaMailSenderImpl е конкретна имплементация на JavaMailSender интерфейса.

Конфигурацията на JavaMailSenderImpl включва настройки като име на хоста, потребителско име, парола и порт за изпращане на електронни съобщения.

Тези настройки се взимат от стойностите, дефинирани в анотациите @Value. Също така се задават и други настройки като протокол за транспортиране на пощата, автентификация, стартов TLS протокол и дебъг информация.

Всички тези настройки се задават в Properties обект, който се връща от метода getJavaMailProperties() на обекта JavaMailSenderImpl.

```
@Configuration  
public class MailConfig {  
    @Value("${spring.mail.host}")  
    private String host;  
    @Value("${spring.mail.username}")  
    private String username;  
    @Value("${spring.mail.password}")  
    private String password;  
    @Value("${spring.mail.port}")  
    private int port;  
    @Bean  
    public JavaMailSender mailSender() {  
        JavaMailSenderImpl sender = new JavaMailSenderImpl();  
        sender.setHost(host);  
        sender.setPort(port);  
    }  
}
```

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”

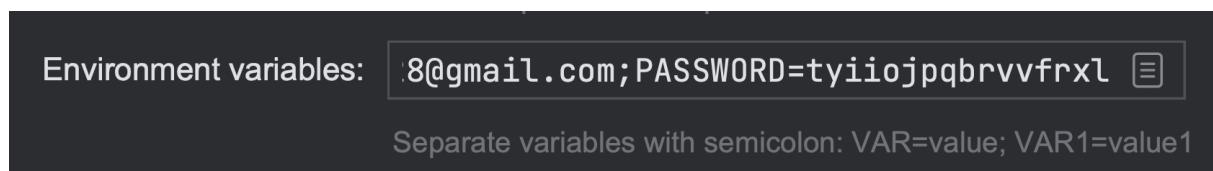
ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

```
    sender.setUsername(username);
    sender.setPassword(password);

    Properties props = sender.getJavaMailProperties();
    props.put("mail.transport.protocol", "smtp");
    props.put("mail.smtp.auth", "true");
    props.put("mail.smtp.starttls.enable", "true");
    props.put("mail.debug", "true");
    return sender;
}

}
```

Потребителското име и паролата се вземат от „променливите на средата“, които сме ги дефинирали в интеграционната среда . Това се прави с цел сигурност,ако някой има достъп до нашия код когато е публикуван в интернет,той няма да може да види имейла и паролата,понеже са дефинирани само за нас в нашата машина.



Фигура 4-3 Снимка,която показва „променливите на средата“ на бек-енда

След това имаме папка „security_configuration“ конфигурация за аутентикацията и авторизацията на приложението

Той определя какви са правата на достъп за различните уеб адреси (URL) в приложението.

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

```
.authorizeRequests().antMatchers("/v1/user/admin/**")
.hasRole("ADMIN")

.antMatchers("/v1/user/")
.permitAll()

.antMatchers("/v1/user/**")
.hasAnyRole("ADMIN", "USER")



.antMatchers("/v1/advertisement/admin/**")
.hasRole("ADMIN")

.antMatchers("/v1/advertisement/all/**")
.permitAll()

.antMatchers("/v1/advertisement/**")
.hasAnyRole("ADMIN", "USER")



.antMatchers("/v1/category/admin/**")
.hasRole("ADMIN")

.antMatchers("/v1/category/categories/**")
.permitAll()

.antMatchers(HttpMethod.GET, "/v1/image/**")
.permitAll()

.antMatchers("/v1/image/**")
.hasAnyRole("ADMIN", "USER")




.antMatchers("/v1/email/sendmail/")
.hasRole("ADMIN")

.antMatchers("/v1/userdetails/**")
.hasAnyRole("ADMIN", "USER")
```

Следват набори от .antMatchers(), където всеки такъв метод се отнася до определен URL адрес или група от адреси. За всяка група адреси се задава набор задава набор от права за достъп.

Където има .hasAnyRole означава, че потребителят трябва да има една от тези роли (Админ или обикновен потребител в този случай), за да има достъп до тази страница.

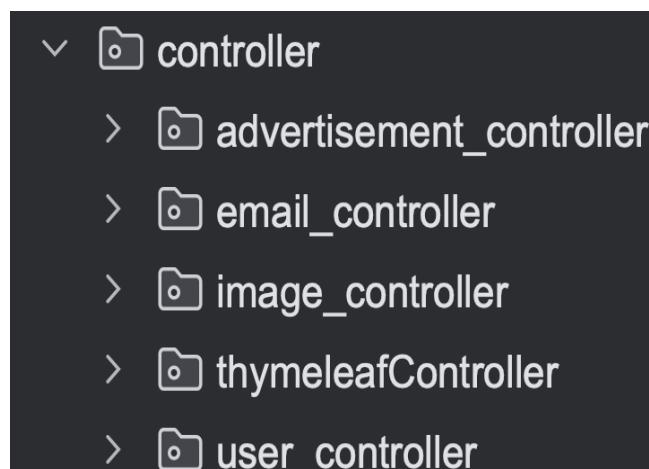
ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

Където има само .hasRole(Роля) там трябва да съответната роля, за да има достъп до тази страница.

Където има .permitAll() означава, че тази страница е достъпна за всички (за нерегистрираните, администратора и обикновените потребители)

В папката „thymeleaf_configuration“ е конфигурацията ни за thymeleaf.

В папката „controller“ и там има класове, които обработват HTTP заявки от клиента. Контролерите в Spring Boot са отговорни за обработката на заявките, които пристигат от клиента и връщането на отговор на клиента.



Фигура 4-4 Снимка, която показва файловата структура на контролера

Ето ви част от един контролер:

```
@RestController
@RequestMapping("/v1/user")
public class UserController {

    private final UserService userService;

    public UserController(UserService userService) {
        this.userService = userService;
    }

    @GetMapping("admin/getAllUsers")
    public ResponseEntity<List<UserDto>> getAllUsers() {
        return ResponseEntity.ok(userService.getAllUsers());
    }
}
```

Анотацията "@RestController" означава, че този клас е контролер и същевременно връща резултатите като RESTful услуги.

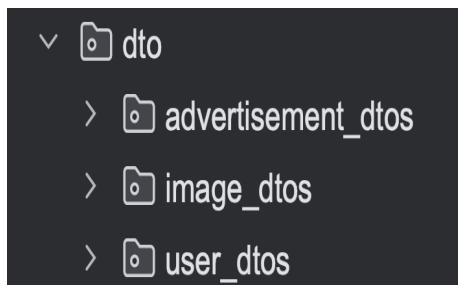
Анотацията "@RequestMapping" служи за мапиране на HTTP заявки към методи от контролера. В този случай всички HTTP заявки, които започват с "/v1/user" ще бъдат обработени от този контролер.

Dependency injection-а на съответния сървис на контролера се прави чрез конструктор.

Имаме метод маркиран с анотацията "@GetMapping". Този метод ще обработи HTTP GET заявката, която има път "/v1/user/admin/getAllUsers". Методът извиква метода "getAllUsers()" от UserService обекта и връща списък от UserDto обекти в HTTP отговора. HTTP отговорът се обвива в ResponseEntity обект, който позволява на контролера да задава статус кода и HTTP заглавието на отговора.

DTO е съкращение от Data Transfer Object (обект за прехвърляне на данни). DTO са обекти, които се използват за прехвърляне на данни между различни слоеве на приложението или между различни микроуслуги в разпределена архитектура. Основната цел на DTO е да капсулират данните, които трябва да бъдат прехвърлени, като по този начин се намалява количеството на прехвърлените данни и се подобрява производителността. DTO обикновено съдържат само необходимите полета и нямат бизнес логика или методи.

И в папката „dto“ имаме такива класове



Фигура 4-5 Снимка на файловата структура на dto-та

Също така тук имаме и заявки, които клиентът може да изпраща, конвертори и валидации.

Заявките(request-тите) са специфичен тип DTO, които представляват входящи HTTP заявки, направени към уеб приложение. Също така тук са и валидациите:

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

Анотацията @NotBlank означава, че това поле не трябва да е празно.

@Pattern

Това е анотация в Java, която се използва за да се проверява дали стойността на полето username отговаря на определен регулярен израз. Ако не отговаря връща грешка.

```
3 usages
@NotBlank
@Pattern(
    regexp = "^[a-zA-Z0-9._-]{3,}$",
    message = "Error username format")
private String username;
```

Фигура 4-6 Снимка, която показва код от валидацията за приложението

Конверторите на DTO са функции или класове, които отговарят за конвертирането между DTO и обекти на домейни.

В папката “enumeration” има класове, които съдържат фиксираны стойности използвани в таблиците в базата данни. Използвани са енъми при ролите съответно в енъма са записани стойностите „ROLE_ADMIN“ и „ROLE_USER“. След това извикваме в ентитатата енъмите, които стават фиксираны стойности в таблиците.

В папката "exception" се намират изключенията, които служат за прихващане на възникнали грешки по време на работа на приложението, като това се осъществява без да спира работата на приложението. Тези изключения се използват за да се прихванат грешките и да се покаже каква е причината за тяхното възникване. В приложението са използвани RuntimeException изключения, защото искаем приложението да продължи да работи, дори когато възникнат грешки, които сме обработили.

В папката „model“ са ни обектите:

Ето част от кода на обекта „User“

```
@Entity
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @Column(unique = true)
```

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

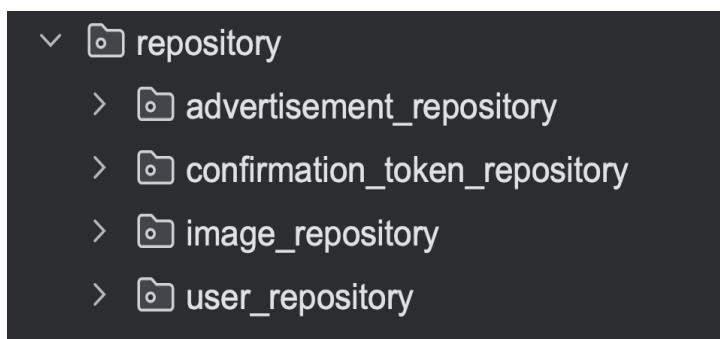
```
private String username;
private String password;
@Enumerated(EnumType.STRING)
private Role role=Role.ROLE_USER;
@Column(unique = true)
private String mail;
```

Този код представлява клас User, който е маркиран с анотацията @Entity, което означава, че той е обектно-релационно отображение (ORM) на таблица в базата данни.

Анотацията @Id служи за маркиране на полето id като основен ключ на таблицата. Полето е с тип Long, което означава, че това е числов стойност, която се увеличава автоматично (auto-increment) при добавяне на нов запис в таблицата, поради анотацията @GeneratedValue със стратегия GenerationType.IDENTITY.

@Column също е анотация, която показва, че това поле съответства на колона в таблицата. В този случай, анотацията се използва за полетата username, password и mail, като се задават различни настройки за всяка една от тях. Например, unique = true означава, че стойностите в колоната трябва да бъдат уникални. @Enumerated(EnumType.STRING) се използва за маркиране на полето role като перечисление (enum) и задава, че стойностите на полето ще бъдат запазени като низови стойности в базата данни, а не като числа. Така този клас User представлява модел на таблица в базата данни, която има колони id, username, password, role и mail. Като анотациите задават различни настройки за всяко едно поле, което да се запази в базата данни.

Папката „repository“, е за да функционира приложението на Spring са нужни репозиторита, които са главната връзка между компонентите в приложението.



Фигура 4-7 Снимка, която показва файловата структура на „repository“

```
@Repository  
public interface UserRepository extends JpaRepository<User, Long> {
```

Фигура 4-8 Снимка, която показва част от код

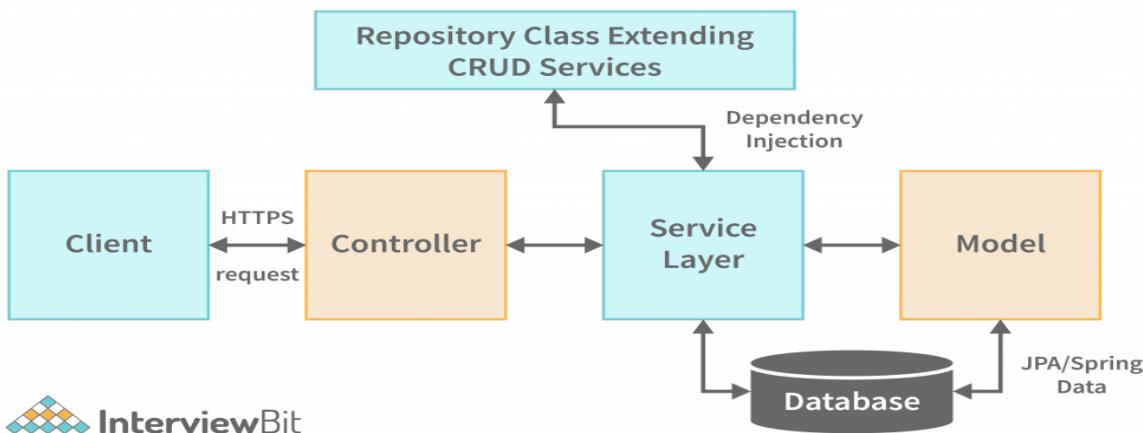
За създаване на репозитория е нужда анотацията “Repository” и наследяването на джава клас JpaRepository, в който се поставят класа, в който е създадена съответната таблица и типът на id-то в базата

В папката „service“ стои бизнес логиката на сайта.

Във всеки клас в папката се съдържат методи, чрез които може да се редактират, изтриват и показват всички обекти и т.н.

Освен това, класът "service" трябва да комуникира само със своята база данни и не трябва да комуникира с "repository" на друг обект. За да комуникира с друга база данни, трябва да комуникира със "service" на другия клас. Това е архитектурата на Spring Boot!

Spring Boot Flow Architecture

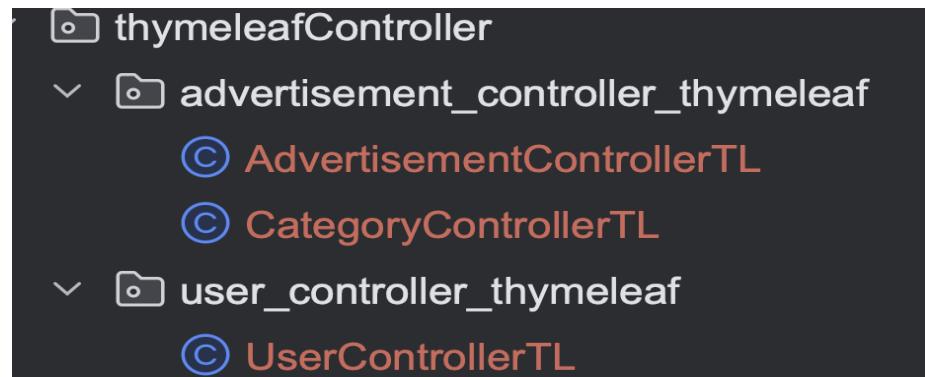


Фигура 4-9 Снимка, която показва архитектурата на SpringBoot

В папката „thymeleafController“ има класове, които в действителност правят връзката между сървисите и интерфейса.

В класа има методи със съответните анотации. Там се оказва в кой темплейт да отиде информацията, която връща съответен метод.

При въвеждане на определен url да се отваря HTML темплейт, в който показваме информацията, която искаме.



Фигура 4-10 Снимка, която показва файловата структура на уеб контролерите

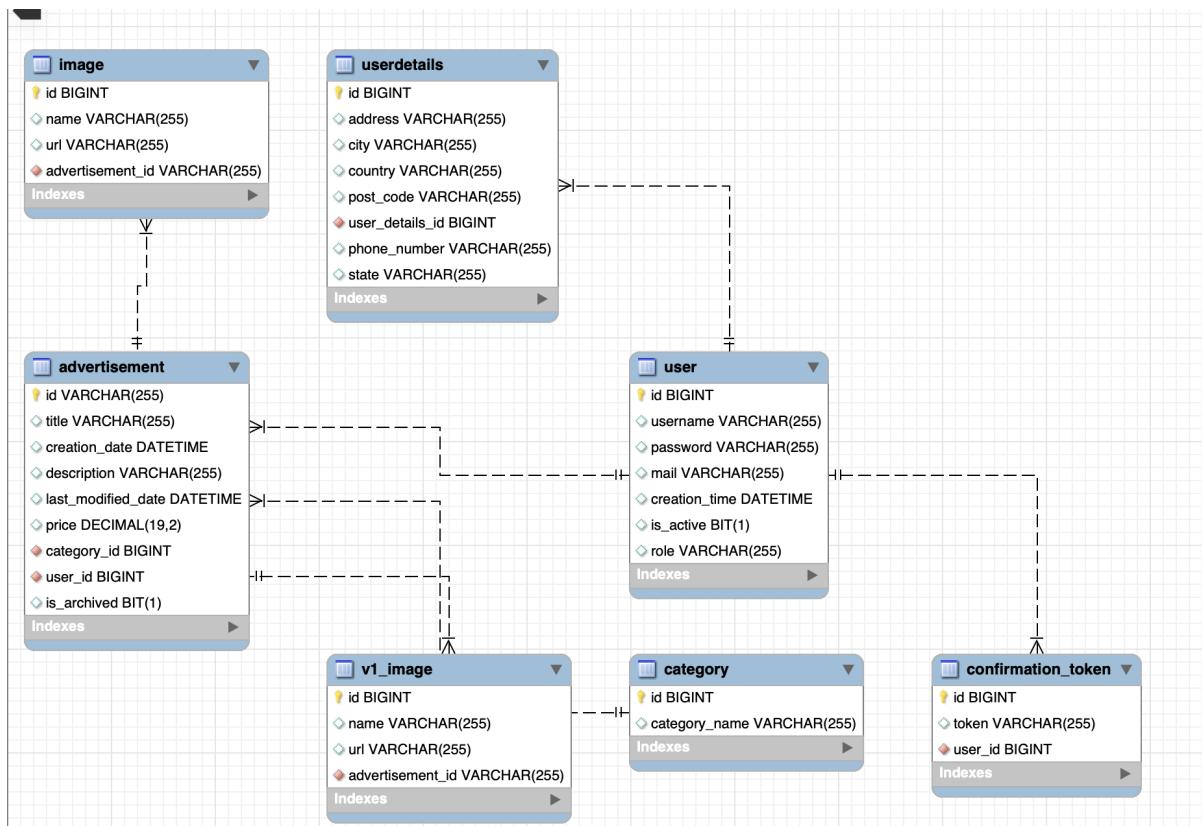
```
@GetMapping("admin/activateUserById/{id}")
public String activateUserById(@PathVariable Long id, Model model) {
    UserDto user=userService.activateUser(id);
    model.addAttribute("user", user);
    return "redirect:/v1/user/admin/showAllActiveUsers";
}
```

Това е метод в Java, който е анотиран с `@GetMapping`, който означава, че той реагира на HTTP GET заявки на съответния URL адрес. В този случай адресът е `"admin/activateUserById/{id}"` - той може да съдържа динамичен параметър, който се подава в URL-то като пътеви параметър, наречен `"id"`. Методът има два параметъра - `"id"`, който е тип `Long` и моделът на данните `"model"`, който е обект, който позволява да се изпращат данни към изгледа на потребителския интерфейс.

Тялото на метода извиква метод на `"userService"`, който е обект, който обработва бизнес логиката на приложението и активира потребител по дадено `"id"`. Резултатът от това извикване е записан в променливата `"user"` от тип `"UserDto"`. След това методът добавя `"user"` към модела на данните и връща редирект към `"showAllActiveUsers"` изгледа, който показва всички активни потребители в системата. Това се постига чрез връщане на стринговият текст `"redirect:/v1/user/admin/showAllActiveUsers"`.

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
 ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

Приложената снимка е схемата на базата данни в приложението. Таблиците със съответните връзки помежду им. Всяко поле е с подходящ тип, който е най-полезен за приложението.



Фигура 4-11 Снимка,която показва Ер диаграмата на таблиците

Във файлът „application.properties“ се съдържат най-важните настройки на базата данни (Паролата и името на базата данни),на имейла(имейл,парола,порт и други). Задава се максималният размер на снимките, които могат да се качат.Също така и къде да се качат. Правят се основни настройки.

```

spring.datasource.url=jdbc:mysql://localhost:3306/beeshop?useSSL=false&serverTimezone=UTC&useLegacyDatetimeCode=false&allowPublicKeyRetrieval=true

spring.datasource.username=root
spring.datasource.password=root

#Hibernate
  
```

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

```
spring.jpa.properties.hibernate.dialect =
org.hibernate.dialect.MySQL5InnoDBDialect

#Hibernate auto ddl
spring.jpa.hibernate.ddl-auto=update

logging.level.org.hibernate.SQL=DEBUG

spring.mail.username=${EMAIL}
spring.mail.password=${PASSWORD}

spring.mail.properties.mail=true

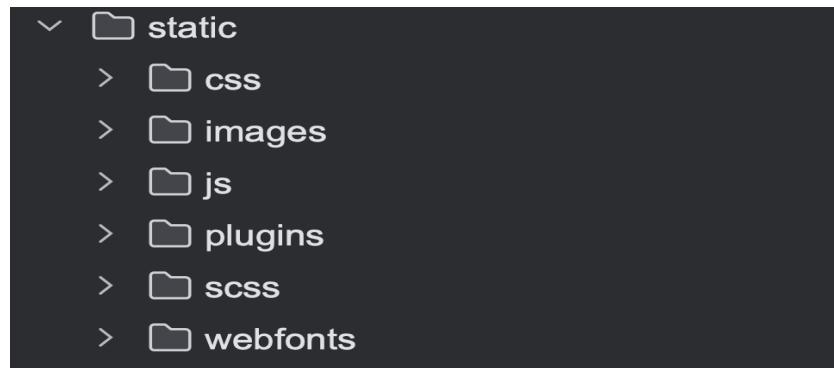
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.properties.mail.transport.protocol=smtp
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true
spring.mail.properties.mail.smtp.starttls.required=true

file.upload-dir=/Users/dzhan/IdeaProjects/to/upload/directory

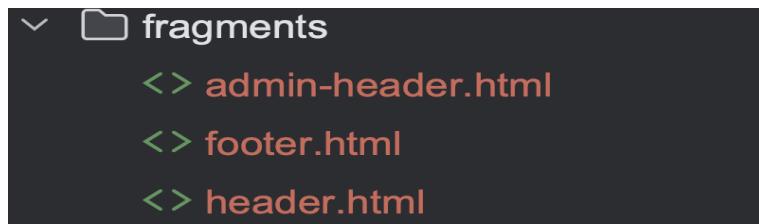
# Set the maximum allowed file size in bytes
spring.servlet.multipart.max-file-size=10MB

# Set the maximum request size (including file data) in bytes
spring.servlet.multipart.max-request-size=10MB
spring.thymeleaf.cache=false
```

В папката “static” има папки ,в които се съдържа стилизацията, снимките и скриптовете



Фигура 4-12 Снимка, която показва файловата структура на уеб частта



Фигура 4-13 Снимка, която показва хедъра и футъра

В папката „templates“ има папка „fragments“ и шаблони на всички страници в приложението. В папката „fragments“ има шаблони за хедър, навигацията и футър. Те се прилагат във всяка една от останалите страници. По този начин не се допуска повтаряне на код. Прилагат се в останалите страници чрез thymeleaf.

```
<html lang="en" xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org"

<header th:replace="fragments/navbar::nav">

<footer class="container-fluid"
        th:replace="fragments/footer::footer"></footer>
```

В темплейтите на страниците има HTML елементи, които създават визията на самия сайт и приложените към всеки елемент класове за стилизация. Примерен код на „advertisements.html“ страницата:

```
<div class="container-fluid products big-padding px-3 bg-honey">
    <div class="container-xl">
        <div class="section-title row">
            <h2 class="fs-1 fw-bold">Всички Обяви</h2>
        </div>
        <div class="row">
            <!-- iterate over the advertisements -->
            <div th:each="advertisement : ${advertisements}" class="col-lg-3
col-md-4 mb-4" >
                <a
                    th:href="@{/v1/advertisement/tl-getAdvertisementById/{id} (id=${advertisement.
advertisementId}) }">
                    <div class="product bg-white p-4 text-center shadow-md">
                        <!-- display the advertisement image -->
                        
                        <!---->

```

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”

ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

```
<div class="d-inline-block">

    <h4 class="fw-bolder fs-4 mt-47"
th:text="${advertisement.title}"></h4>

        <h4 class="fw-bolder fs-5 mt-54"> <span
th:text="${advertisement.userdetails[0].city + '/'
+ advertisement.userdetails[0].state }"></span></h4>

    <h4 class="fw-bolder fs-5 mt-54"> <span
th:text="${advertisement.price+ 'lv'}"></span></h4>

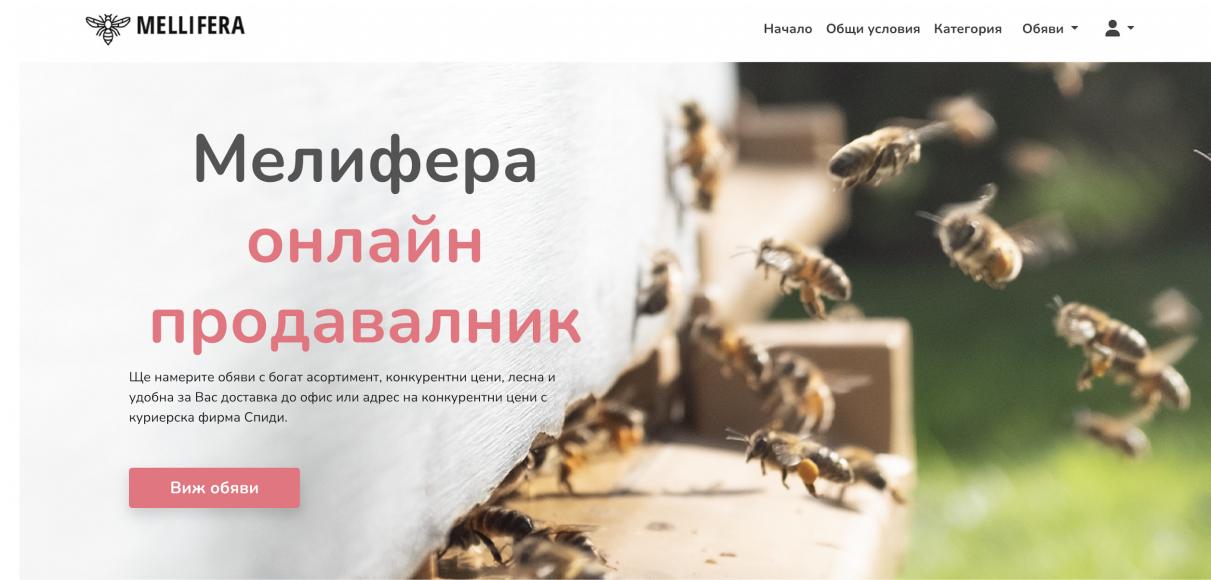
</div>
</div>
```

Глава 5 - ПРИЛОЖЕНИЕ НА ПЧЕЛАРСКА УЕБ ПЛАТФОРМА

При отваряне на сайта се отваря приветстващата страница. В горната част има меню, което съдържа икона за сесия(вход или регистрация) (при вече направена регистрация), общи условия ,категории, обяви.

Отляво може да се види логото на сайта интерактивно създаден, специално за приложението.

В средата има секция, в която кратко е описано приложението. Има един бутона,който ни отвежда при активните обяви на сайта.



Фигура 5-1 Снимка,която показва основната страница на приложението

5.1 Функционалност за регистриране на потребител

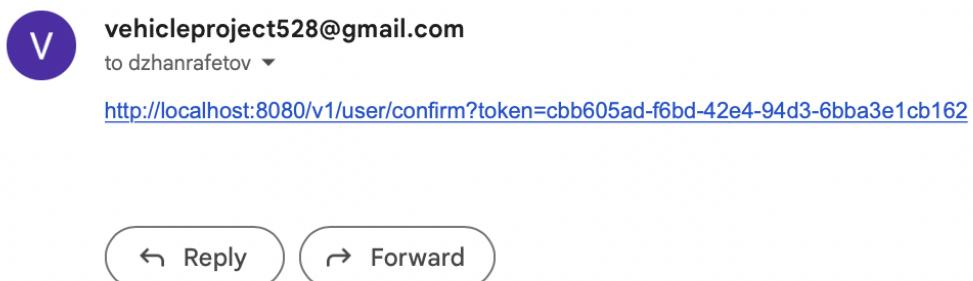
Всеки нов човек, който е заинтересован от идеята на приложението има възможност да създаде профил, чрез който ще може да използва всички функционалности, които предлага приложението на хората, които са се логнали успешно в системата.

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

Във регистрационната форма трябва да се въведат следните данни: потребителско име, парола, имейл и парола. Те трябва да са валидни спрямо изискванията на сайта, в противен случай системата ни връща страница с грешно въведените полета. След като регистрацията е успешна, праща ни на друг формуляр за регистрация.

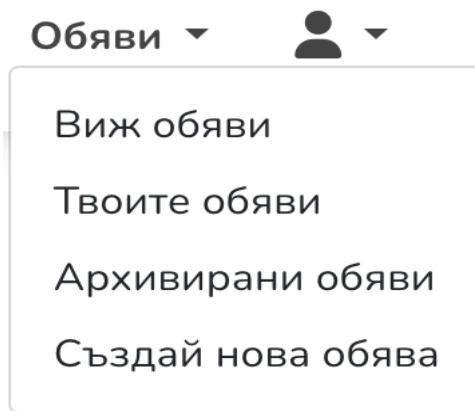
И там трябва да се попълнят следните полета: за адрес, град, пощенски код, държава и телефонен номер. След като и тази операция е успешна. Изпраща се имейл за верификация на този имейл, който сме посочили в първия формуляр.

Натискаме линка, който е на имейла ни, препраща ни в логин формуляра на сайта и акаунтът вече е активиран.



Фигура 5-2 Снимка, която показва линка за потвърждение, изпратен на имейла

Когато сме влезли в системата от навигационното меню отидем с мишката на „ОБЯВИ“ Се появява едно падащо меню.



Фигура 5-3 Снимка, която показва падащото меню на „Обяви“

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

Когато натиснем виж обяви ще ни се появят всичко активни обяви на приложението:



Фигура 5-4 Снимка от сайта, която показва всичките обяви

Когато натиснем върху обявата, ще ни изведе на друга страница за по-подробна информация на обявата:

Категория: Пчелни Кошери

» Начало » Продукти » Пчелни Кошери

Пчелни кошери по НПП

Добавена в: 25/04/2023, 01:18
Описание
Фирма "МАРИАМ 2016 ЕООД"
Цена: 85.00lv

Контакт с продавача
потребителско име: admin
телефон: +359897758062
имейл: novuserr@abv.bg

Фигура 5-5 Снимка, която показва информация за обявата

Когато натиснем от падащото меню „Твоите обяви“, ще покажи всички активни обяви на потребителя:

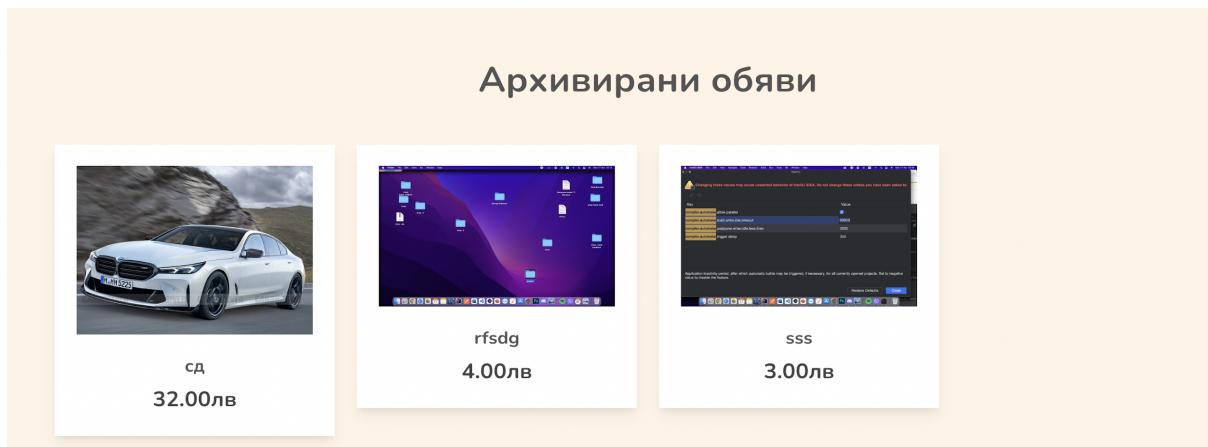
ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН Рафетов 18510



Фигура 5-6 Снимка, която показва активните обяви

Имаме бутон „Архивирай“, когато го натиснем ще архивира обявата и ще се махне от Активни обяви.

Когато натиснем от падащото меню „Архивирани обяви“ ще видим всички архивирани обяви на потребителя.



Фигура 5-7 Снимка, която показва архивираните обяви

Когато натиснем от падащото меню „Създай обява“ ще ни изпрати в страницата, в която ще можем да създаваме обява:

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

Въвеждаме необходимите данни, като спазваме валидациите. И натискаме бутона за създаване на обява.

Създай Обява

Заглавие:

Пчелна Майка Карпатка

Описание:

Оплодени Пчелни Майки! Със ЗАПИСВАНЕ!

Предимството на Карпатските майки е че те са много по-издръжливи през Зимата, по-едри са, снасят повече, носят повече мед и не са агресивни.

Цена:

25

Category:

Пчелни Майки

Снимка:

Choose file

about_04-1024x768.jpeg

Създай Обява

Фигура 5-8 Снимка, която показва формата за създаване на обяви

И ни препраща към обявата, която сме добавили:

Категория: Пчелни Майки

» Начало » Продукти » Пчелни Майки

Пчелна Майка Карпатка

Добавена в: 26/04/2023, 01:16

Описание

Оплодени Пчелни Майки! Със ЗАПИСВАНЕ! Предимството на Карпатските майки е че те са много по-издръжливи през Зимата, по-едри са, снасят повече, носят повече мед и не са агресивни.

Цена: 25.00lv



Контакт с продавача

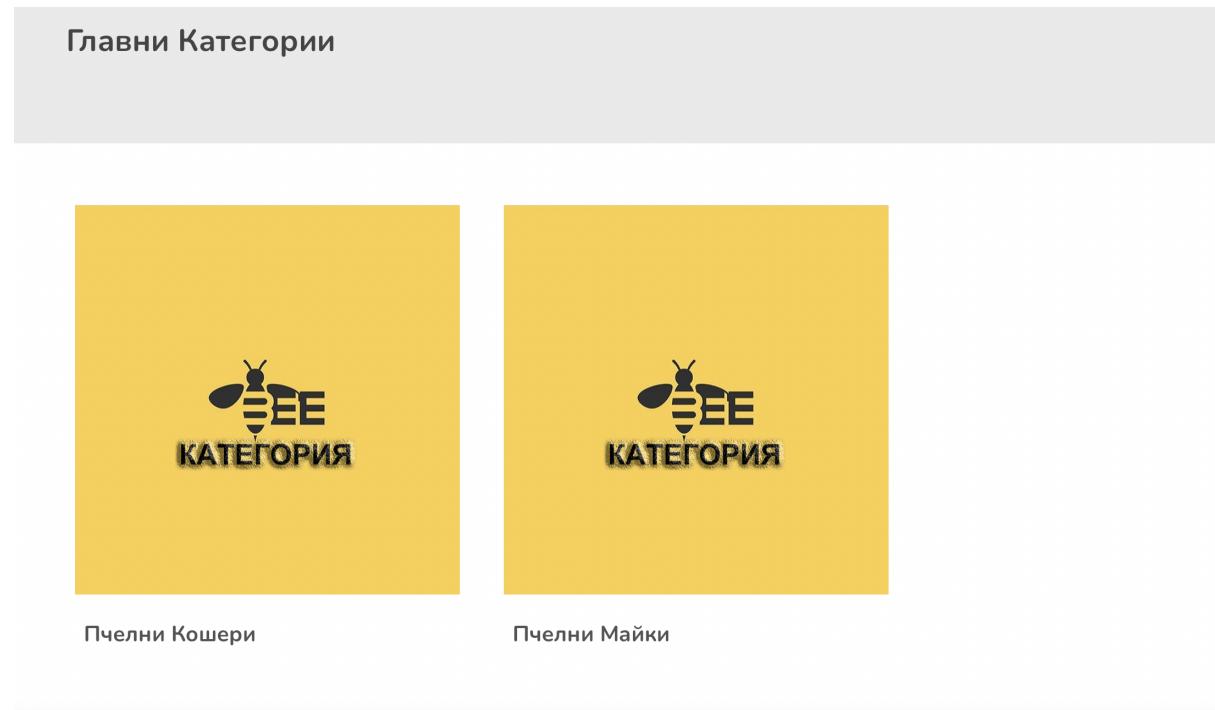
потребителско име: user

телефон: asdfasd

Фигура 5-9 Снимка, която показва обявата, която сме добавили

ДИПЛОМНА РАБОТА: „Разработка на пчеларска уеб платформа“
ДИПЛОМАНТ: ДЖАН Рафетов 18510

Когато натиснем от навигационното меню „Категория“ ще видим главните категории на сайта:



Фигура 5-10 Снимка,която показва категориите на сайта

Ако искаме да видим активните обяви само от избрана категория,натискаме върху категорията,на която искаме да видим.

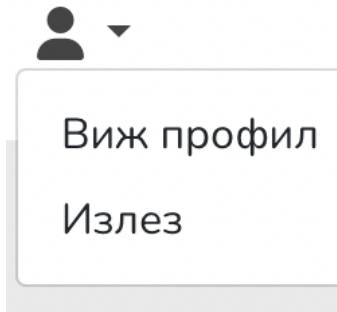
Активни обяви от категория : Пчелни Кошери



Фигура 5-11 Снимка,която показва обявите от дадена категория

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

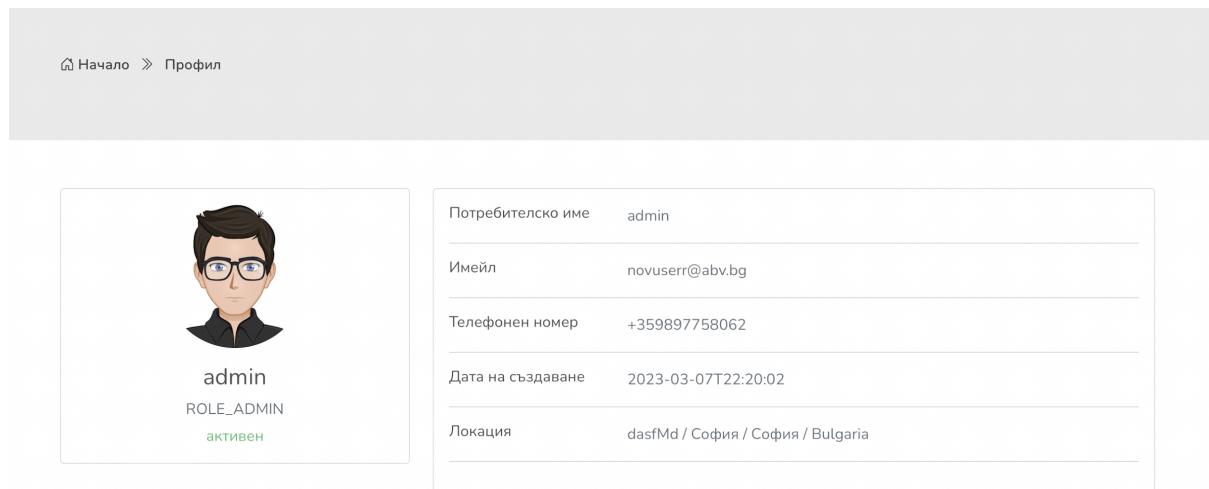
Когато отидем с курсора върху иконата(човече) в навигационното меню, ще ни излезе следното падащо меню:



Фигура 5-12 Снимка, която показва падащото меню на профил

И когато натиснем „Виж профил“

Ще видим данните на профила ни:



| | |
|-------------------|-----------------------------------|
| Потребителско име | admin |
| Имейл | novuserr@abv.bg |
| Телефонен номер | +359897758062 |
| Дата на създаване | 2023-03-07T22:20:02 |
| Локация | dasfMd / София / София / Bulgaria |

Фигура 5-13 Снимка, която показва информация за профила, в който сме

Когато натиснем „Излез“, ще излезем от профила ни и ще ни препрати в логин формуляра.

Когато влезем като администратор ни приветства този начален еcran, можем да го наречем Статистическо Табло:

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510



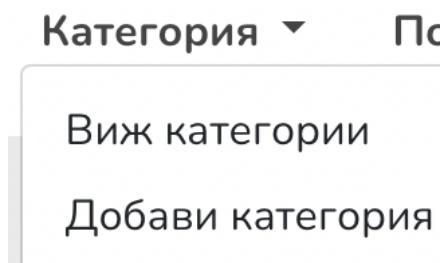
Фигура 5-14 Снимка, която показва основната страница на администратора

Също така като влезем като администратор имаме друго навигационно меню:



Фигура 5-15 Снимка, която показва хедъра на администратора

От падащото меню на „Категория“ ни излиза това:



Фигура 5-16 Снимка, която показва падащото меню на „Категория“

като натиснем „Виж категории“

Ни излизат всички категории на сайта и можем да изтриваме категории.

Категории



Пчелни Кошери

Изтрий Категория



Пчелни Майки

Изтрий Категория

Фигура 5-17 Снимка, която показва категориите на сайта

Като натиснем „Добави Категория“ ни препраща към следната страница:

Създай Категория

Име на категория:

Създай Категория

Фигура 5-18 Снимка, която показва формата за създаване на категория

Откъдето можем да добавяме нови категории.

Когато отидем с курсора върху „Потребители“, ще ни излезе:

Потребители ▾ Обяви ▾

Активни Потребители

Архивирани Потребители

Фигура 5-19 Снимка, която показва подаците от менюта „Потребители“

Когато натиснем „Активни Потребители“ ще ни препрати към следната страница

| ид: | потребителско име: | статус: | емей: | действие |
|-----|--------------------|---------|---------------------|-------------|
| :26 | user | активен | canrafet0@gmail.com | Deактивирай |
| :27 | admin | активен | novuserr@abv.bg | Deактивирай |

Фигура 5-20 Снимка, която показва активните потребители

Където можем и да деактивираме потребители

А когато натиснем „Архивирани Потребители“ ще можем да видим всички архивирани потребители:

Deактивирани потребители



ПОТРЕБИТЕЛИ

ид: 64

потребителско име: nov432

неактивен

имейл: dzhansrafet3ov@gmail.com

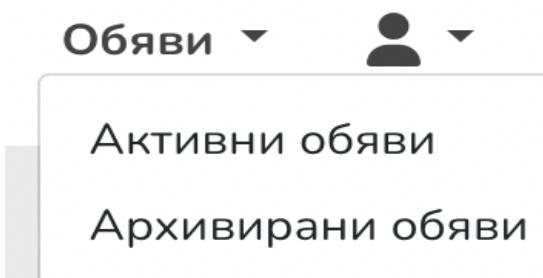
Активирай

Изтрий

Фигура 5-21 Снимка, която показва деактивираните потребители

Където можем да ги активираме или да ги изтрием от сайта.

Когато отидем с курсора върху „Обяви“ ще ни излезе следното:



Фигура 5-22 Снимка, която показва падащото меню на „Обяви“

Когато натиснем „Активни обяви“ ще видим всички активни обяви на сайта

ДИПЛОМНА РАБОТА: “Разработка на пчеларска уеб платформа”
ДИПЛОМАНТ: ДЖАН Рафетов 18510

Активни Обяви



Пчелни кошери по НПП
София/София
85.00лв

[Деактивирай Обява](#)



Пчелна Майка Карпатка
Правец/София
25.00лв

[Деактивирай Обява](#)

Фигура 5-23 Снимка, която показва активните обяви

Където можем да деактивираме обявата.

Когато натиснем от падащото меню „Архивирани обяви“ ще видим всички архивирани обяви в сайта.

Архивирани обяви



сд
Zlatna panega/Sofia
32.00лв

[Активирай Обява](#)

[Изтрий Обява](#)



probvame
София/София
32.00лв

[Активирай Обява](#)

[Изтрий Обява](#)



rfsdg
Zlatna panega/Sofia
4.00лв

[Активирай Обява](#)

[Изтрий Обява](#)



Spring Boot – How to reload the changes without restarting the server?
София/София
43.00лв

[Активирай Обява](#)

[Изтрий Обява](#)

Фигура 5-24 Снимка, която показва архивираните обяви

Където ще можем да ги активираме или да ги изтрием.

Глава 6 - ИЗВОДИ И ЗАКЛЮЧЕНИЕ

Сайтът може да се превърне в основен източник на полезна информация за българското интернет пространство, като се фокусира върху определена група хора. Мнението на потребителите може да обогати и развие сайта. Приложението има здрава основа, която може да се развива допълнително. Едно от най-важните подобрения е добавянето на търсачка, за да се улесни навигацията на потребителите. Редактирането на профили е също много важно и администраторът трябва да има право да редактира всеки един от тях. Потребителите трябва да имат възможност да сменят своите пароли и имена на профилите си. Допълнително, опция за комуникация(чат) може да се добави, за да се подобри връзката между купувачите и продавачите.

ДИПЛОМНА РАБОТА: “Разработка на пчеларска учебна платформа”
ДИПЛОМАНТ: ДЖАН РАФЕТОВ 18510

Глава 7 - ИЗПОЛЗВАНА ЛИТЕРАТУРА

1. <https://www.youtube.com/@Java.Brainz>
2. <https://www.javatpoint.com/spring-boot-architecture>
3. <https://www.geeksforgeeks.org/spring-boot-architecture/>
4. <https://spring.io>
5. <https://classroom.google.com/>
6. <https://github.com/>
7. <https://www.javabrains.io/>
8. <https://www.youtube.com/c/FolksDev>

Глава 8 - СПИСЪК С ИЗПОЛЗВАННИТЕ ОЗНАЧЕНИЯ И СЪКРАЩЕНИЯ.

HTML – Hyper Text Markup Language

CSS - Cascading Style Sheets

SQL - Structured Query Language

HTTPS - Hypertext Transfer Protocol Secure

URL - Uniform Resource Locator

OOP - Object-oriented programming

MVC – model view controller

SMTP- Simple Mail Transfer Protocol

DTO - Data Transfer Protocol

JPA- Java Persistence API

ORM - Object-Relational Mapping