**НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАЦИОННИ ТЕХНОЛОГИИ**

**2023/2024 УЧЕБНА ГОДИНА**

**Проект № 49**

**Картина, която съдържа символ, лого, текст, Шрифт

Описанието е генерирано автоматично**

**Both Sides of the Coin**

Автори: Емил Дианов Петров,

Цветослав Анатолиев Ангелов

ППМГ „Академик Иван Ценов“

гр. Враца

**Съдържание**

1. Тема ................................................................................................................................... 3
2. Автори ............................................................................................................................... 3
3. Ръководители .................................................................................................................... 3
4. Резюме ............................................................................................................................... 4
   1. Цели ............................................................................................................................. 4
   2. Основни етапи в реализирането на проекта ............................................................ 4
   3. Ниво на сложността на проекта ................................................................................ 5
   4. Логическо и функционално описание на решението ............................................. 6
   5. Реализация ................................................................................................................ 13
   6. Описание на приложението .................................................................................... 14
   7. Заключение ............................................................................................................... 21
5. **Тема**

* Онлайн платформа за покупко-продажба на продукти между търговци и клиенти.

1. **Автори**

* Емил Дианов Петров
* ЕГН: 054702
* Адрес: област Враца, община Мездра, г. Мездра, ул. „Христо Смирненски“ №1
* Телефон: 0877804105
* Имейл адрес: [timkaq67@gmail.com](mailto:timkaq67@gmail.com)
* Училище: ППМГ „Академик Иван Ценов“, гр. Враца
* Клас: 12
* Цветослав Анатолиев Ангелов
* ЕГН: 054108
* Адрес: област Враца, община Мездра, с. Руска Бела, ул. „Георги Димитров“ №24
* Телефон 0895164680
* Имейл адрес: [cveti3060@gmail.com](mailto:cveti3060@gmail.com)
* Училище: ППМГ „Академик Иван Ценов“, гр. Враца
* Клас: 12

1. **Ръководители**

* Станислава Каменова
* Телефон: 0879003288
* Имейл адрес: [kamenova@bitex.bg](mailto:kamenova@bitex.bg)
* Длъжност: старши учител по информатика и ИТ
* Албена Цеков
  + Телефон: 0888407298
  + Имейл адрес: [albena2008@abv.bg](mailto:albena2008@abv.bg)
  + Длъжност: старши учител по информатика и ИТ

1. **Резюме**
   1. **Цели**

Изработеният проект представлява уеб приложение, служещо за улесняване на търговските сделки между продавачи и купувачи. Целта на приложението е да създаде удобна и ефективна платформа, където потребителите могат да публикуват свои обяви за продажба на различни продукти или услуги, както и да разглеждат и закупуват предложенията на други потребители. Основната цел на проекта е да предостави интуитивен и лесен за използване интерфейс, който да улесни процесите на търговия, като намали времето и усилията, необходими за намиране и закупуване на желаните продукти. Като цяло, приложението се стреми да създаде позитивно и удовлетворително изживяване както за продавачите, така и за купувачите, като им предостави сигурно и надеждно средство за онлайн търговия.

* 1. **Основни етапи в реализирането на проекта**
* Дефиниране на целите и нужните функционалности, избор на тема
* Създаване, проектиране на база от данни, нанасяне на редакции
* Имплементиране на бизнес логика и нейното коригирането
* Визуално оформление на потребителската и административната част
* Валидиране на правотата на изпълнението
* Отстраняване на грешки
* Създаване на рекламни материали
* Публикуване, разпространение
* Итеративен модел на работа, повтаряне на определени стъпки в процеса на разработка, при нужда
  1. **Ниво на сложност на проекта**

Сложността на проекта се определя от сървърната (бизнес) логика, както и от потребителската (визуализираща) логика. Главна цел е коректната работа на описаните части, добавянето на профил, разполагащ с различни нива на достъп (контролен профил, следящ за неподходящи обяви).

Основен модел в създаването на проекта беше имплементирането на трислойна архитектура (Model-View-Controller – MVC), нейното разширяване да включва и Service layer – т. нар. Бизнес логика. В процеса на разработка е използван свободно разпространяващ се проектен шаблон.

Друг основен момент беше разпределението на потребителите в различни роли,

ограничаването на възможностите на неоторизираните потребители да променят съдържанието и обратното – потребителите с определени роли (продавач, купувач) да имат по-големи права и възможности за редакция или действие.

* 1. **Логическо и функционално описание на решението**

Проектът използва за основа трислойния модел – MVC. Добавен е слой за бизнес логиката – т. нар. Service layer.

* Презентационният (view) слой е изграден от HTML5, Tailwindcss, JavaScript, Blade templating engine.
* Контролиращият (controller) слой е свързващо звено в MVC, той определя каква функционалност (бизнес логика) да извършена, гарантира правото на потребителя да извършва дадено действие, връща на потребителя очаквания резултат (view), а при нужда – и съобщение за грешка.
* Бизнес (service, business logic) слоят използва езикът PHP и работна рамка Laravel Core за работа с данни, генериране на съдържание, определяне на правата за достъп и други.
* Слоят от данни (data access layer, model) се грижи за информацията, нейното съхранение, обновяване, изтриване и достъпване. Използвана е MySQL релационна система за управление на бази данни, достъпът до нея се осъществява чрез технологията Laravel.

Чрез Tailwindcss се постига гъвкав (responsive) дизайн, който прави приложението подходящо за достъпване от най-различни устройства – от мобилни телефони, таблети, до настолни компютри. Използвана е външна библиотека позволяваща връзката със Stripe.

* **Създаване и управление на клиентски профили:** Позволява на

приложението да създава и управлява профили на клиенти, които

могат да се свържат с техни кредитни или дебитни карти.

* **Обработка на плащания:** Позволява на приложението да извършва

различни видове плащания, като еднократни плащания, абонаменти,

връщане на средства и други.

* **Сигурност на данните:** Stripe предоставя високо ниво на сигурност

при обработката на плащания, като възприема най-новите стандарти

за защита на данните и сигурност на транзакциите.

* **Анализ на транзакциите:** Позволява на приложението да проследява

и анализира транзакциите, включително данни за продажби, брой

успешни транзакции и други метрики.

* **Интеграция с различни платежни методи:** Stripe поддържа

множество платежни методи, включително кредитни и дебитни

карти, мобилни портфейли и алтернативни платежни методи.

Разглеждайки Laravel, като популярен PHP фреймуърк, той включва

различни мерки за сигурност, които защитават от обичайни атаки срещу уеб

приложения. Някой от защитите и функциите за сигурност, предоставени от Laravel,

включват:

* **Защита от CSRF** атаки **(Cross-Site Request Forgery):** Laravel автоматично

Генерира CSRF токени за всяка активна потребителска сесия. Тези

токени се използват за проверка дали удостовереният потребител е

този, който прави заявките към приложението.

* **Защита от** **XSS** атаки **(Cross-Site Scripting):** Blade шаблонният двигател на

Laravel автомати чно избягва изхода по подразбиране, което помага за

предотвратяване на XSS атаки. Освен това, Laravel предоставя функции

като директивата **`{{ }}`**, която показва сурови неекранирани данни

при необходимост.

* **Предотвратяване на SQL Инжекции**: Eloquent ORM на Laravel по

подразбиране използва параметризирани заявки, което помага за

предотвратяване на SQL инжекции. Също така предоставя Query Builder,

който автоматично ескейпва входните параметри.

* **Аутентикация и Оторизация**: Laravel предлага надеждна система за

аутентикация по подразбиране, включително функции като сигурно

хеширане на пароли, криптиране и аутентикационни драйвери.

* **Валидация на Входните Данни**: Laravel предоставя удобни методи за

валидиране на входни данни на потребителя, което помага за

предотвратяване на обичайни сигурностни уязвимости като инжекции.

* **Сесионна Сигурност**: Laravel позволява на разработчиците да

конфигурират настройките за сесията, включително опции за

криптиране и сигурни бисквитки, за подобряване на сесионната

сигурност.

* **Ограничаване на Скоростта и Дърпането**: Laravel предоставя вградена

поддръжка за ограничаване на скоростта и дърпането, за да се

предпази от атаки с брутална сила и атаки на Отказ на Обслужване

(DoS).

* **Сигурни Заглавия**: Разработчиците могат лесно да конфигурират

сигурни заглавия като Политика за Сигурност на Съдържанието (CSP),

Строга Сигурност на Транспортния Протокол (HSTS) и Споделяне на

Ресурси от Различен Произход (CORS), за да подобрят сигурността.

* **Сигурност при Качване на Файлове**: Laravel предоставя функции за

сигурно качване на файлове, включително валидация на файлове,

проверка на MIME типове и конфигурация на местоположението за

съхранение.

* **Контейнер за Внедряване на Зависимости**: Контейнерът за внедряване

на зависимости на Laravel помага за управлението на зависимостите по

сигурен начин, намалявайки риска от уязвимости като инжекции на код.

Приложението използва уеб сървър <https://jnhost.net/> за хостинг

и за отваряне

по подходящ начин на потребителските заявки.

Създаването на база данни в Laravel се извършва чрез използването

на миграции и модели. Миграциите са PHP файлове, които описват

структурата на таблиците и техните взаимоотношения. Те позволяват

автоматизирано създаване и промяна на базата данни. Моделите са

PHP класове, свързани с таблиците в базата данни, които предоставят

методи за взаимодействие с данните. Чрез тях можем да извършваме

операции като извличане, създаване, обновяване и изтриване на

записи в базата данни. Използването на миграции и модели позволява

лесно управление на структурата и данните в базата данни в Laravel,

което е едно от най-важните неща при разработката на уеб

приложения (предстои променяне на таблици от базата данни).

* Картина, която съдържа текст, екранна снимка, Шрифт

  Описанието е генерирано автоматичноКартина, която съдържа текст, екранна снимка, Шрифт, номер

  Описанието е генерирано автоматичноКартина, която съдържа текст, екранна снимка, номер, дисплей

  Описанието е генерирано автоматичноEER диаграма на базата данни

Софтуерните файлове са разпределени в следните папки:

* Controllers – файлът съдържа цялата бизнес логика
* Models – съдържат се Eloquent ОRM Models
* Database – папката съдържа миграции за създаване на таблиците
* Картина, която съдържа текст, екранна снимка

  Описанието е генерирано автоматичноКартина, която съдържа текст, екранна снимка

  Описанието е генерирано автоматичноRoutes –

Картина, която съдържа текст, екранна снимка

Описанието е генерирано автоматично

* Storage – съхранява всички снимки на продукти
* View – папката съдържа Blade templates, които отговарят за изгледа на приложението
  1. **Реализация**

Използвани са следните технологии и инструменти за реализацията на проекта:

* HTML5 – маркиране на елементите по страниците и тяхното отделяне едни от други
* Tailwind CSS – стилизиране и дизайн, визуално оформление
* Blade template – дизайн тема на проект
* Flowbite icons – икони и визуално оформление
* JavaScript – анимации и функционалност в браузъра
* PHP, Laravel framework – бизнес логика, MVC архитектура
* MySQL – релационна система за управление на бази данни
* JNHost – уеб сървър
* Visual Studio Code – визуален редактор на текст
  1. **Описание на приложението**

Приложението е достъпно на уеб адрес: [https://r.notohost.com/bsc](https://r.notohost.com/bsc" \t "_blank" \o "https://r.notohost.com/bsc) , като то може да бъде достъпвано от модерни уеб браузъри, например Firefox, Chrome, Edge, Opera GX и техните мобилни версии.

Потребителите могат да бъдат гости или регистрирани потребители. Акаунтите на регистрираните потребители могат да имат следните роли:

* Администратор (Admin)
* Продавач (Seller)
* Купувач (Buyer, Client)

В зависимост от ролята, която има даден акаунт, са налични различни нива на достъп и правомощия.

Основни функционалности на приложението „Both Sides of the Coin“ включват:

* Регистрация и управление на потребители:
  + Потребителите могат да създават профил в системата като продавач и купувач.
  + Те могат да преминават от купувач към продавач
  + Възможност за вход и изход от акаунта.
* Публикуване на обяви от продавачите:
  + Продавачите могат да създават и управляват обяви за своите продукти или услуги
  + Добавяне на снимки, описание, цена и други детайли към обявите.
* Търсене и филтриране на обяви от купувачите:
  + Купувачите могат да преглеждат и търсят обяви, които са публикувани от продавачите.
* Добавяне на продукти в количка и покупка:
  + Купувачите могат да добавят желаните продукти в количката и да ги закупят
* Управление на профил:
  + Потребителите могат да управляват своите профили, включително редактиране на лична информация, промяна на парола и други настройки.

Инструкции за инсталиране под Windows ОС:

1. Инсталиране на необходимите компоненти:

* Инсталиране на РНР: Изтеглете и инсталирайте РНР от официалния уебсайт. След инсталирането, добавете пътя до РНР в системната променлива РАТН.
* Инсталиране на Composer: Изтеглете и инсталирайте Composer от официалния уебсайт. След инсталацията, уверете се, че пътят към него е добавен към системната променлива РАТН.
* Инсталиране на Visual Studio Code: Изтеглете инсталационния файл за него от официалния уебсайт и следвайте инструкциите на екрана, за да завършите процеса на инсталация.

1. Инсталиране на Laravel:

* Отворете командния ред (Command Prompt) и изпълнете командата

`**composer global require laravel/installer**`.

1. Инсталиране на Tailwind CSS:
   * В директорията на вашия проект, отворете терминал и изпълнете командата

`**npm install tailwindcss**`, за да инсталирате Tailwind CSS.

1. Инсталиране на разширения за Visual Studio Code:
   * Отворете Visual Studio Code и инсталирайте следните разширения чрез секцията „Extensions“: Laravel Blade Snippets и GitLens.
2. Инсталиране на проекта от GitHub:
   * Клонирайте кода на проекта от GitHub хранилището на адрес [https://github.com/Proto69/OnlineMarket](https://github.com/Proto69/OnlineMarket" \t "_new), като използвате следната команда в терминала:

`**git clone** [https://github.com/Proto69/OnlineMarket`](https://github.com/Proto69/OnlineMarket%60)

1. Инсталиране на зависимости:
   * Отворете терминал в директорията на проекта и изпълнете командата `**composer install**` за инсталиране на всички зависимости, описани във файла `**composer.json**`.
2. Конфигуриране на средата на разработка:
   * Създайте копие на файлът `**env.example**` и го кръстете на `**.env**`.
   * Настройте вашият файл `**.env**` със съответните настройки за база данни, сесии и други настройки според вашите нужди.
3. Създаване на ключ за приложението
   * Отворете терминал и изпълнете командата `**php artisan key:generate**`, за да генерирате уникален ключ за вашето Laravel приложение.
4. Стартиране на локалния сървър:
   * Изпълнете командата `**php artisan serve**`, за да стартирате вградения локален съврър на Laravel.
   * При промяна на класовете в HTML кода се препоръчва изпълнение на командата `**npm run build**`, за да се свалят потенциално добавените свойства.

Подробно ръководство е поместено в потребителската документация.

На началната страница на приложението потребителите виждат съобщение с информация за проекта, насоки за използване.

* + **Администратор (Admin)**

Ролята, отговаряща за наблюдение на правилната употреба на

потребителските профили, включва набор от функционалности, осигуряващи

ефективен контрол и надзор върху дейността на потребителите в системата. Освен

основната задача за банване на обяви с неподходящо съдържание, тази роля

разполага също така и с редица други функционалности:

Ролята може да изтрие продукт, ако е определен като неподходящ за

платформата, създавайки по този начин сигурна и етична общност.

При повторни нарушения от страна на потребителския акаунт, ролята може

да предприеме мерки за изтриване на акаунта, гарантирайки спазване на

правилата и дисциплина в общността.

Ролята има възможността да проследи историята на наложените наказания

за всички акаунти в системата, което позволява ефективно управление на

дисциплинарните мерки и предотвратяване на нарушенията.

Освен това, ролята може да преглежда всички продукти и акаунти в

системата, за да осигури контрол и наблюдение на активностите в платформата.

Ролята разполага с функционалност за търсене на акаунт по име, което

улеснява намирането на конкретни потребители в системата.

В случай на обжалвания или отмяна на нарушението, ролята може да

отмени наложеното наказание на акаунт или продукт.

Ролята е отговорна и за управлението на обжалванията, представени от

потребителите, като анализира и решава нарушенията съгласно правилата и

процедурите на системата.

1. Начало на формуляра
2. Край на формуляра
   * **Продавач (Seller)**

Продавачът разполага със специализиран потребителски профил, който му

предоставя удобството да управлява своите продукти и да следи тяхната

продажба. Следователно, функционалностите, налични за продавача, включват:

При вход в системата, продавачът има възможност да създаде или да влезе

в своя акаунт в Stripe, където може да управлява финансовите си транзакции и да

следи плащанията от купувачите.

Продавачът може да работи в тестов режим без създаване на акаунт в

Stripe, което му позволява да тества функционалностите на платформата преди да

активира истински плащания.

Създаване на продукт е лесно и удобно за продавача, като може да добавя

снимка и други характеристики, които описват продукта му.

Продавачът има възможност да активира или деактивира продуктите си

според наличността или други фактори.

Промяна на характеристиките на продуктите е възможна по всяко време,

като продавачът може да обнови информацията за своите продукти според

променящите се условия или изисквания.

Историята на продадените продукти предоставя на продавача подробна

информация за всяко закупено от него стоки, включително данни за купувача за

доставка и връзка, което му помага да управлява и да следи своите продажби.

Продавачът може да преглежда всички свои продукти едновременно,

осигурявайки си бърз достъп до информацията за тях и лесно управление на целия

си каталог.

* Купувач (Buyer, client)

Купувачът разполага със специален потребителски профил, който му

предоставя лесен и удобен начин за пазаруване и управление на поръчките си.

Функционалностите, налични за купувача, включват:

Купувачът може да търси продукти по име, което му позволява бързо и

лесно да намери желаните стоки сред голямото разнообразие на платформата.

Той има възможност да купува продукти от различни търговци в рамките на

една поръчка, което му осигурява гъвкавост и удобство при пазаруването.

Плащането за всички продукти от различни търговци се осъществява чрез

Stripe checkout, който гарантира бързо и сигурно плащане за всички артикули в

поръчката.

Историята на поръчките предоставя на купувача подробна информация за

предишните му покупки, включително статуса на поръчката и информация за

доставката.

Купувачът може да плаща неплатени поръчки и да ги завършва,

осигурявайки си бърз и лесен начин за управление на финансовите си транзакции.

Ако е необходимо, купувачът има възможност да поправя неплатена

поръчка, като добавя или премахва продукти според своите нужди.

Той може също така да изтрие неплатена поръчка, ако реши, че вече не

желае да продължи с нея, осигурявайки си лесен начин за отмена на ненужни

поръчки и управление на своя акаунт.

* 1. **Заключение**

В заключение, проектът "Both Sides of the Coin" представлява иновативно уеб приложение, което дава възможност на потребителите да се включат в онлайн търговията с удобство и лекота. Чрез интуитивен интерфейс и богат набор от функционалности, приложението предоставя на продавачите платформа за лесно управление на своите продукти и поръчки, както и възможност за проследяване на техните транзакции. За купувачите, "Both Sides of the Coin" предлага гъвкавост и удобство при пазаруването, позволявайки им да намират и пазаруват продукти от различни търговци с едно плащане. С интеграцията на Stripe за обработка на плащанията, приложението гарантира бързо и сигурно пазаруване за всички потребители. С обхвата си на функционалности и удобството, което предлага за потребителите, проектът "Both Sides of the Coin" е предназначен да преобрази начина, по който хората пазаруват и продават онлайн, като създаде благоприятна и иновативна среда за електронна търговия.

Има няколко функционалности, които са в процес на разработка: статистики за продавача, изпращане на имейл при успешна транзакция и получаване на наказание, филтриране на продукти, рейтинг на услуги, няколко вида езици, мобилно и десктоп приложение за допълнително удобство на клиентите.