СЕТУ “ Михајло Пупин” – Скопје

ПРОЕКТНА ЗАДАЧА
ПО ВЕБ ПРОГРАМИРАЊЕ

ТЕМА: Симулација на банкарски трансакции

Ментор:
Проф.Поликсена Митева

Изработил:
Берат Ахметај

Скопје, март 2021

Содржина:

Вовед

Истражувањето на “Институт за развој на електронски комуникации” покажува делумно разочарувачки бројки, во Македонија само 2 од 14 банки имаат мобилна апликација која овозможува безготовинско плаќање преку NFC стандардот, додека пак две помали банки воопшто немаат мобилна апликација.

Овој проект е симулација на веб-апликација за е-банкарство, при што безбедноста и оптимизацијата на апликацијата беа приоритет број еден при развојот на овој проект.

Почнувајќи од главната страница, сè до поставките на корисникот, сè се испорачува со користење на стандардни безбедносни програмски услуги, единственото нешто што се симулира во оваа апликација се трансакциските монети.

Оваа веб-страница е структурирана и оптимизирана за да биде и веб-апликација, ја користи најновата технологија „PWA“ (прогресивна веб апликација) која ја трансформира веб-страницата и сите нејзини услуги да се користат на сите уреди (Десктоп компјутери, Мобилни уреди, Смарт телевизори, па дури и Смарт фрижидери).

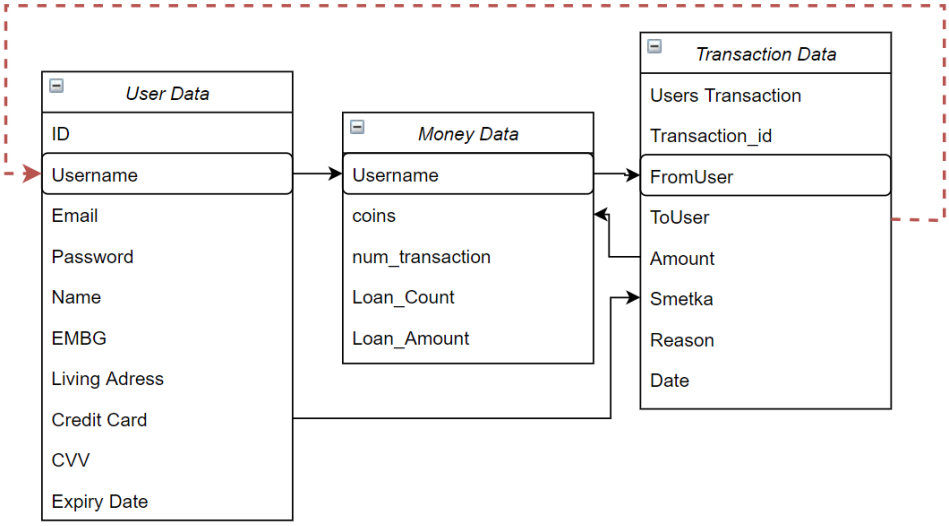
Инспириран од уметникот Кендрик Ламар, ја именував оваа симулирана апликација за е-банкарство како „MoneyTree“ и целата идеја е околу управувањето со парите лесно, како да растат на дрвја. Има повеќе услуги во рамките на веб-апликацијата, секој корисник може да депонира/врши трансакции, да гледа историја на трансакции, преглед на статистики, промена на деталите за нивниот профил и употреба на калкулатор специјално направен за банкарство. Сето ова е претставен преку едноставен кориснички интерфејс.

Ќе започнеме со анализа на некои техники за тоа како е направена веб-апликацијата, неговото структурирање, како работи безбедноста и како се чуваат трансакциите во базата на податоци што корисникот може да ја гледа во секое време, на било кој уред. И на крајот ќе го разгледаме брендирањето на проектот, дизајнот, маркетингот, корисничкиот интерфејс и корисничкото искуство на целиот проект.

Структурата На Податоци

Датотечниот систем е составен од 3 табели кои складираат податоци дадени од корисникот:

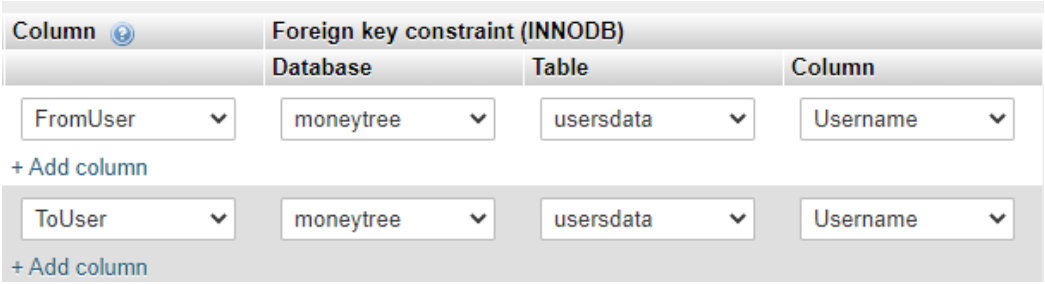
1. Кориснички податоци
2. Податоци за пари
3. Податоци за трансакции



Слика 1.0 - Дијаграм на сите податоци од базата на податоци на апликацијата

Базата на податоци за овој проект користи 3F нормализација, што значи дека користи надворешни клучеви за поврзување на ентитетие во табелите и осигурување дека нема дуплирани податоци, што го прави безбеден и ефикасен начин за складирање на податоци за е-банкарство.

Пример за како работи структурата на податоците: Во момент кога корисникот прави трансакција до друг корисник на апликацијата, првин се врши анализа на код (се објаснува понатаму) и ако се верифицираат двете корисници, тогаш во базата на податоци се вметнуваат сите податоци за трансакцијата, но и автоматски двете корисници се поврзани во базата, со што администраторот но и програмата може полесно да ги бара ако треба манипулирање на историја трансакции или уредување на податоците. Кажано технички, ова се прави со помош на надворешни клучеви (foreign keys) и користење на релациони табели (relational tables)



Слика 1.1 - Слика на како се вметнати надворешните клучеви во базата на податоци

Безбедноста на податоците

Безбедноста заслужува цела страница на објаснување, затоа што создавањето апликација за е-банкарство треба да биде една од најсигурните апликации, како што беше речено, овој проект беше структуриран околу безбедноста на податоците на корисниците.

Еден од начините на кои ги обезбедуваме податоците е со криптирање на сè што корисникот испраќа на серверот, во овој пример ќе ви ги дадам моите вистински информации за најавување што ги користам за да се најавам на оваа веб-апликација (MoneyTree)

Email	Pass	Username
beratahmetaj02@gmail.com	$2y$10$Oqlo26yNJOW0mL/l0hG59eYglKhgkkOlHB.FTgppehM...	berat

Слика 2.0 - Слика направена од административната страна на базата на податоци, може да се видат моите персонални информации за логирање на вебстраната

Сега кога ви ја дадов мојата е-пошта, лозинка и корисничко име, можеби ќе се обидете да се најавите во веб-апликацијата, но нема успеете, затоа што лозинката е шифрирана, само корисникот ја знае својата лозинка, вие не, ниту јас, иако ги имам сите информации за корисниците зачувани на мојот сервер, тоа е затоа што со една линија код се криптираат низа на знаци, во специфичен шифриран код од 60 карактери.

```
$hashedpass = password_hash($pass,PASSWORD_DEFAULT);
```

Слика 2.1 - Линија код која ја претвара лозинката во енкриптирана лозинка

Користење Сесии

Веб сесија е серија од податоци кои се чуваат во интернет пребарувачот(Browser) на корисникот и серверот.

Овој проект користи сесии за да го следи корисникот додека тој манипулира со неговите податоци, користиме сесии за да следи кој е најавен и да следи колку монети има корисникот и сите негови податоци за влез на платформата.

```php
<?php
session_start();
//стартаме сесија во страната
//Го зимаме корисничкото име и го ставаме во сесија
//Во овој случак корисничкото име е земено од момент кога
//корисниот е логиран се до крај на одлогирање
$username = $_SESSION['username'];
```

Слика 3.0 - Линија код која дефинира сесија во веб страната

login_action.php

```php
<?php

if (isset($_POST["submit"])) {

    //старт на сесија
    session_start();
    //ги ставаме податоците од POST во SESSION за да ги користиме
$_SESSION['username'] = $_POST["username"];
$_SESSION['pass'] = $_POST["pass"];

//Зимаме од POST во обичен variable за да ги користиме во функции
    $username = $_POST["username"];
    $pass = $_POST["pass"];
```

Слика 3.1 - Линија код која дефинира сесија во веб страната

На двете слики погоре можеме да видиме session_start() што е функција што ја започнува сесијата на веб-страницата на која е корисникот, и на двете слики можеме да видиме дека корисничкото име е донесено за да се користи на веб-страницата.

logout_action.php

```php
<?php
//Ги добиваме сите податоци од сесијата
$_SESSION = array();

// Ги бришиме сите податоци од сесијата
if( ini_get( "session.use_cookies" ) ) {
    $params = session_get_cookie_params();

    setcookie(
      session_name()
      , ''
      , time() - 42000
      , $params[ "path"     ]
      , $params[ "domain"   ]
      , $params[ "secure"   ]
      , $params[ "httponly" ]
    );
}

// и на крај го бришаме и самата сесија, со цел да нема траг од податоците
if( session_status() === PHP_SESSION_ACTIVE ) { session_destroy();}
header("Location: ./login.php?error=Logout");
?>
```

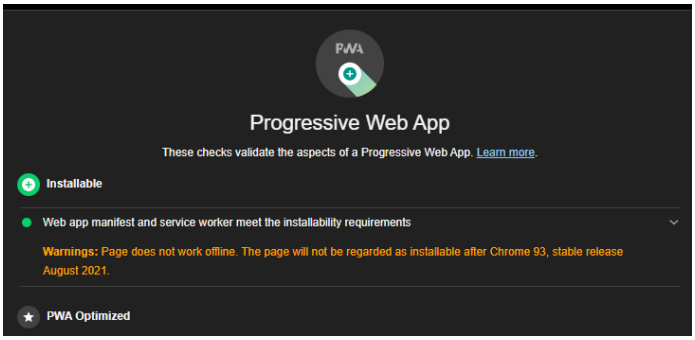
Слика 3.2 - програмски код за одлогирање

На ист начин како што ги користиме сесиите за да го најавуваме корисникот, ни треба и начин да го одјавиме корисникот и да ја избришеме сесијата, тоа го правиме преку кодот прикажан на слика 3.2 кој што исто така вклучува и објаснување на користените функции и методи во коментарите.

Едно нешто што треба да се забележи е дека има код „time() - 42000“ што значи дека сесијата се зачувува во прегледувачот(browser) на корисниците 4200 секунди или 1 час, што значи дека корисникот може да ја прелистува веб-страницата без да треба да се најавува на секоја секунда, му даваме на корисникот време 1 час за да ја користи платформата, после тоа корисникот се одјавува автоматски. Тоа се користи за безбедносни причини, на пример, ако корисникот ја заборавил својата банкарска сметка на компјутерот, не сакаме некој да ја прелистува неговата сметка, па автоматично го одјавуваме.

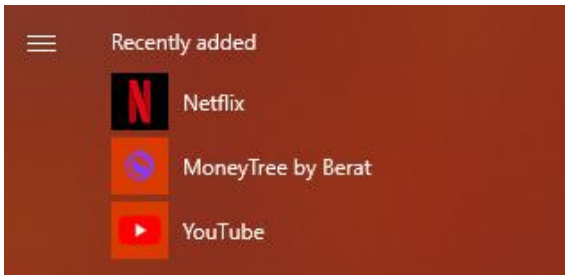
Правење апликација од веб страна

Како што рековме погоре, оваа страница ја користи PWA технологијата за да ја направи оваа веб-страница во веб-апликација, оваа технологија е направена од Google и бара некои стандарди пред да се направи страница во веб-апликација.

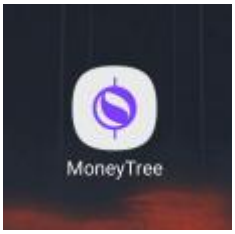


Слика 4.0 - Анализа на вебстраната за дали се квалифицира како веб апликација

На сликата 4.0 можеме да видиме како се прави анализата со користење на Google Chrome Lightouse, резултатите што се враќаат велат дека веб-страницата може да се инсталира на кој било уред, тоа се прави со помош на дефинирање манифест. (Документ што дефинира како ќе работи веб-апликацијата, името на апликацијата, темата, ако има реклами или не, и иконите што се користат за телефони и други мали уреди.)



Слика 4.2 - Апликацијата инсталирана во компјутер



Слика 4.3 - Апликацијата инсталирана на мобилен уред

Сервисен работник е вид на веб-работник (Web Service Worker). Во суштина, тоа е JavaScript-датотека што работи одделно од главната нишка на прелистувачот, пресретнувајќи ги мрежните барања, кеширајќи или враќајќи ги ресурсите од кешот и доставувајќи пораки за push нотификации. Ова дава потенцијал за веб апликацијата да праќа нотификации до корисникот, да може да се отвара offline (кога корисникот не е уклучен во мрежа) но исто така е есенцијален за претварање на веб страната во веб апликација.

```
JS serviceWorker.js > ...
1   const MoneyTree = "Moneytree"
2   const assets = [
3     "/",
4     "/index.html"
5   ]
6
7   self.addEventListener("install", installEvent => {
8     installEvent.waitUntil(
9       caches.open(MoneyTree).then(cache => {
10        cache.addAll(assets)
11      })
12    )
13  })
14
15  self.addEventListener("fetch", fetchEvent => {
16      fetchEvent.respondWith(
17        caches.match(fetchEvent.request).then(res => {
18          return res || fetch(fetchEvent.request)
19        })
20      )
21  })
```

Слика 4.5 - Дефинирање на сервисен работник кој ќе уредува со веб страната за мобилни уреди

На слика 4.5 кодот му дава упатства на интернет прегледувачот да го зачува кешот на сите слики и веб-страницата подготвена за корисникот да ја прелистува, ја подобрува брзината на веб-страницата за 300%.

Кориснички интерфејс и дизајн

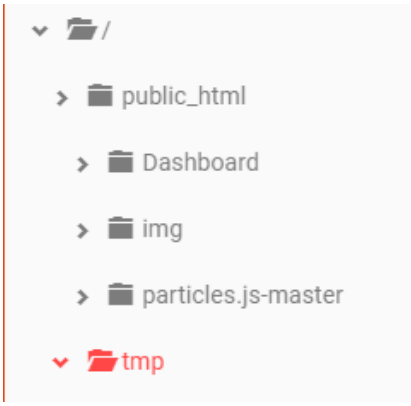
Дизајнот започна како еден од првите чекори за реализирање на оваа веб-апликација, едноставен кориснички интерфејс го привлекува секој корисник, особено кога темата е комплицирана, веб-страниците за е-банкарство можат да бидат сложени, но важно е да биде едноставна за корисниците.

Слика 5.0 - Дизајн Принципи на веб апликацијата

Иако постојат технологии кои го олеснуваат дизајнот со користење на веќе изработени стилови, програмерот може само да ги смени боите, важно е да се создаде уникатно чувство за секоја веб-страница, затоа секој код за стил во оваа веб-апликација беше рачно напишано од мене.

Хостирање на веб апликацијата

Хостирањето е последната тема која ќе ја разгледаме пред да разгледаме како се прават трансакциите и важно е да имате изглед и чувство на целото искуство со веб-апликацијата пред да продолжиме напред. Но пред тоа да објасниме и за хостирањето...



Слика 6.0 - Структурата на фајловите во платформата 000webhost.com

На слика 6.0 е прикажан датотечниот систем и сите негови папки што се хостирани на платформата, тоа е местото каде што веб-страницата и сите нејзини датотеки се чуваат, но тоа не е важно, она што е важно е црвената папка напишана како „tmp“ (temporary/привремено).

Во таа папка се зачувани сите податоци на сесијата, како што рековме погоре, ние користиме сесии за да ги следиме корисниците на веб-апликацијата, откако корисникот ќе се одјави од веб-апликацијата, сесијата е избришана и уништена, но додека корисникот е најавен, сесиите се зачувуваат на серверот, тоа е најсигурен начин за корисниците да ги чуват нивните логинг кредицијали за најавување.

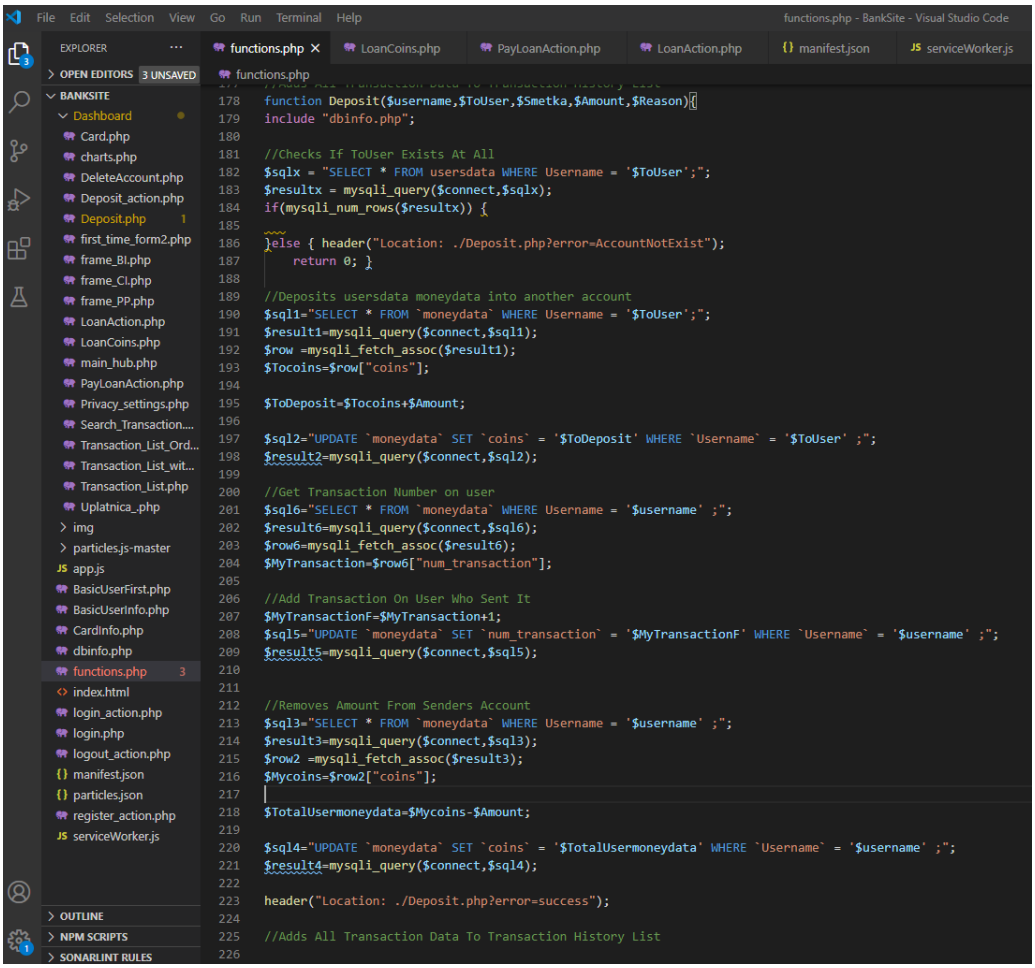
Сега кога ја знаете структурата на веб-страницата и како се чува на Интернет, можете да ја прегледате веб-апликацијата преку овој линк - https://moneytree-berat.000webhostapp.com/.

Логирање и Регистрација на Корисник

Кога корисникот се регистрира, веб-страницата поминува низ некои функции во кодот за прво да провери некои безбедносни проверки. Проверува дали корисничкото име е преземено од друг корисник, дали е-поштата постои во базата на податоци, дали лозинката е силна, и други проверки.Кога корисникот се регистрира, страницата автоматски отвора колони и редови подготвени за новиот корисник, дури и ако не ги знаеме сите информации за новите корисници.

Вршење трансакции во платформата

Иако веб-страницата содржи многу одлики и многу програмски кодови, ќе ја разгледаме најуникатната функција.

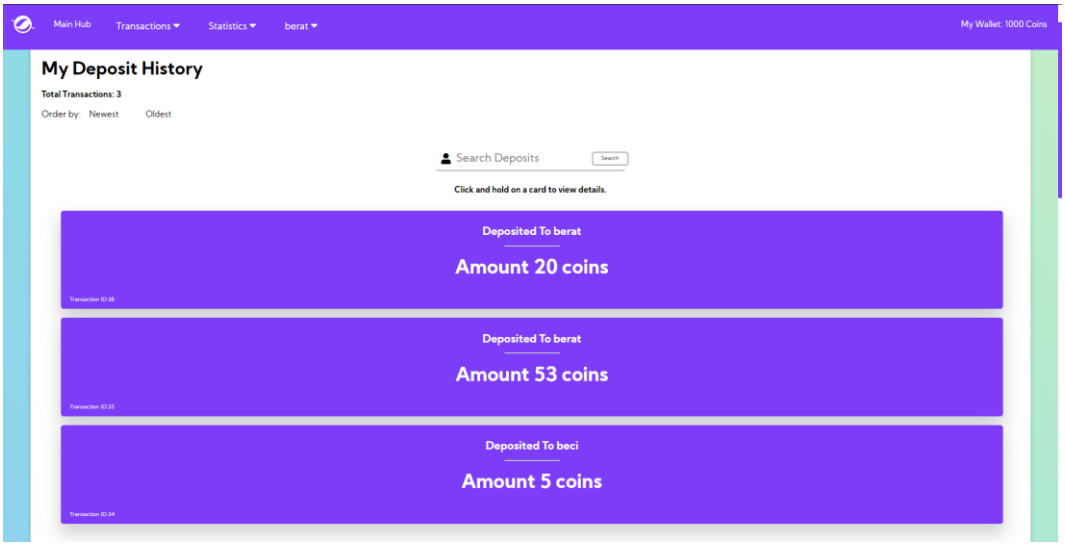


Слика 7.0 - Целиот код за функцијата вршење трансакција

Оваа функција ги проверува сите безбедносни проверки за да провери дали корисникот за кој сакаме да ги испратиме парите постои или не, дали имаме доволно пари за испраќање...

Откако ќе бидат проверени сите безбедносни проверки, функцијата ги депонира парите со тоа што прво ќе ги извади од корисникот и ќе ги испрати на другиот корисник, а на крајот од сето ова ќе ги запише сите детали на низа информации, детали како кој на кого му ги испратил парите, кога и зошто, со цел за корисникот да има историја од трансакции или листа на информации за трансакции.

Листа Трансакции



Слика 8.1 - Кориснички интерфејс за Историја на трансакции

Веб-апликацијата им нуди на корисниците да ги прегледуваат сите нивни трансакции, како што е апликацијата со користење на структурата на податоците што ги објаснивме на првите страници, таа чува историја на трансакции на базата на податоци и е подготвена да му се даде на корисникот.

Корисникот може да манипулира со нивните трансакции со избирање на опаѓачки или растечки редослед или дури и пребарување на специфични трансакции. Сето ова е направено на комплексен начин со користење на врските со базата на податоци и “Front-End” (клиентска страна) јазик за програмирање Javascript за да се прикажат сите информации точно и безбедно.

Заклучок

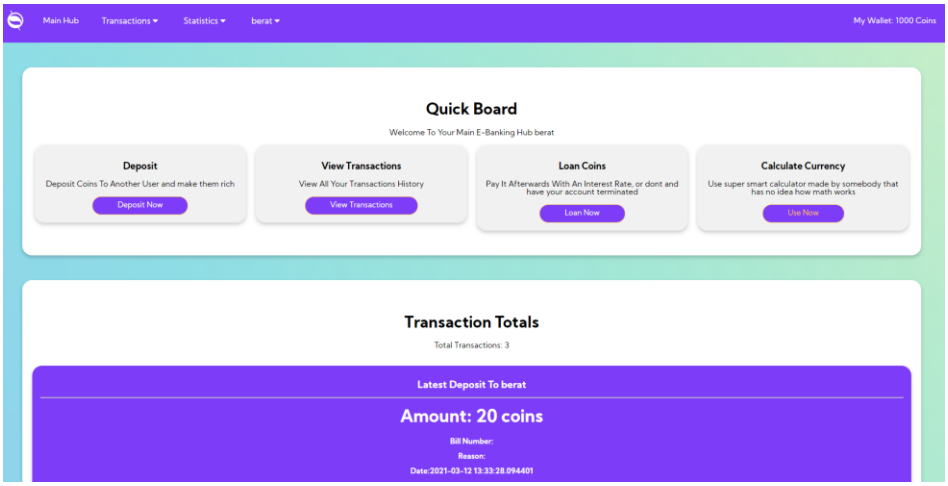
Овој проект е комбинација од 6.930 редови код напишани рачно (да ,ги изброив сите), 30 слики, 3 видеа и безброј часови поминати.

И покрај тоа што звучи многу, уште повеќе е придобивката од знаењето, кога започнав со проектот немав знаење за поврзување на бази на податоци со веб-страници, не знаев што е веб-апликација и не знаев како да програмирам во PHP и да користам Javascript Библиотеки.

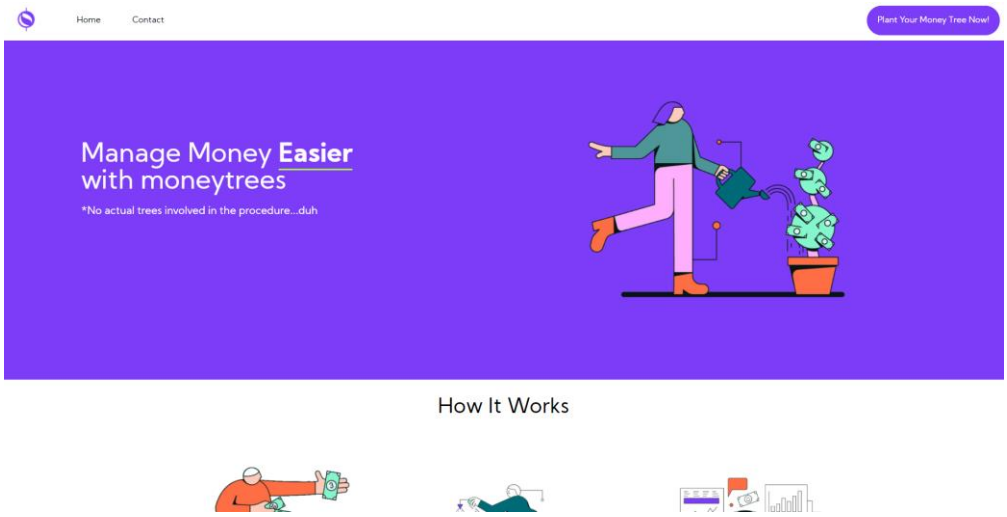
Овој документ може да биде огромен со информации, Но тоа е затоа што е тешко да се стават 6.930 редови код на 10 страници документ. Но, она што се надевам е дека секој што чита ќе добие чувство за овој проект.

Ќе продолжам со ажурирање на веб-страницата исто така ќе продолжам да правам други проекти со кои ќе учам уште повеќе.

Додека го изработував овој проект не престанав да ја слушам песната „MoneyTrees“ од Кендрик Ламар, Таа е песната што ме инспирираше да го започнам целиот овој проект.



Слика 9.0 - Кориснички интерфејс на веб апликацијата



Слика 9.1 - Кориснички интерфејс на веб апликацијата

Можете да ја прегледате веб-апликацијата преку овој линк:
https://moneytree-berat.000webhostapp.com/

Користена литература:

1. **https://inrekom.org.mk/e-bankarstvo-vo-makedonija/**
2. https://web.dev/what-are-pwas/
3. https://hazelcast.com/glossary/web-session/
4. https://developers.google.com/web/tools/lighthouse
5. https://developers.google.com/web/ilt/pwa/introduction-to-service-worker
6. https://en.wikipedia.org/wiki/Style_sheet_(web_development)
7. https://moneytree-berat.000webhostapp.com/