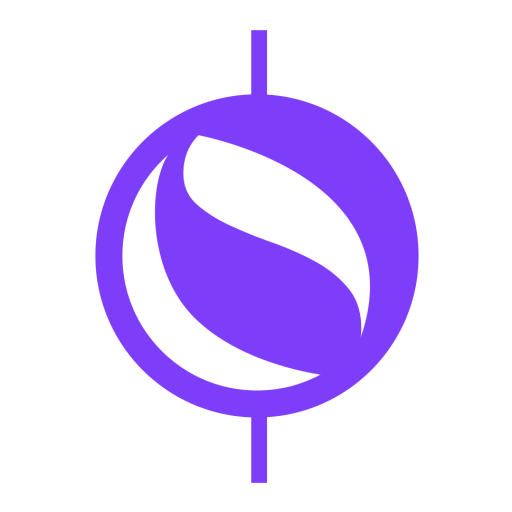
****

**SETU "Mihajlo Pupin" - Skopje**

|  |
| --- |
| **PROJECT WORK**  **WEB PROGRAMMING** |

**Banking Transaction Simulation**

****

Skopje, March 2021

# Introduction

The research of the "Institute for Development of Electronic Communications" shows partly disappointing figures, in Macedonia only 2 out of 14 banks have a mobile application that allows non-cash payment through the NFC standard, while two smaller banks do not have a mobile application at all.

This project is a simulation of a web application for e-banking, where the security and optimization of the application were the number one priority in the development of this project.

From the main page to the user settings, everything is delivered using standard security software services, the only thing that is simulated in this application are transaction coins.

This website is structured and optimized to be a web application, uses the latest technology "PWA" (progressive web application) that transforms the website and all its services to be used on all devices (Desktop computers, Mobile devices , Smart TVs, and even Smart Refrigerators).

Inspired by artist Kendrick Lamar, I named this simulated e-banking app "MoneyTree" and the whole idea is about managing money as easily as growing a tree. There are several services within the web application, each user can deposit / perform transactions, view transaction history, view statistics, change their account details and use a calculator specially made for banking. All this is presented through a simple user interface.

We will start by analyzing some techniques on how the web application is made, its structuring, how security works and how transactions are stored in the database that the user can view at any time on any device. Finally we will look at the project branding, design, marketing, user interface and user experience of the whole project.

# Data Structure

The file system consists of 3 tables that store data given by the user:

1. User data

2. Data on money

3. Transaction data

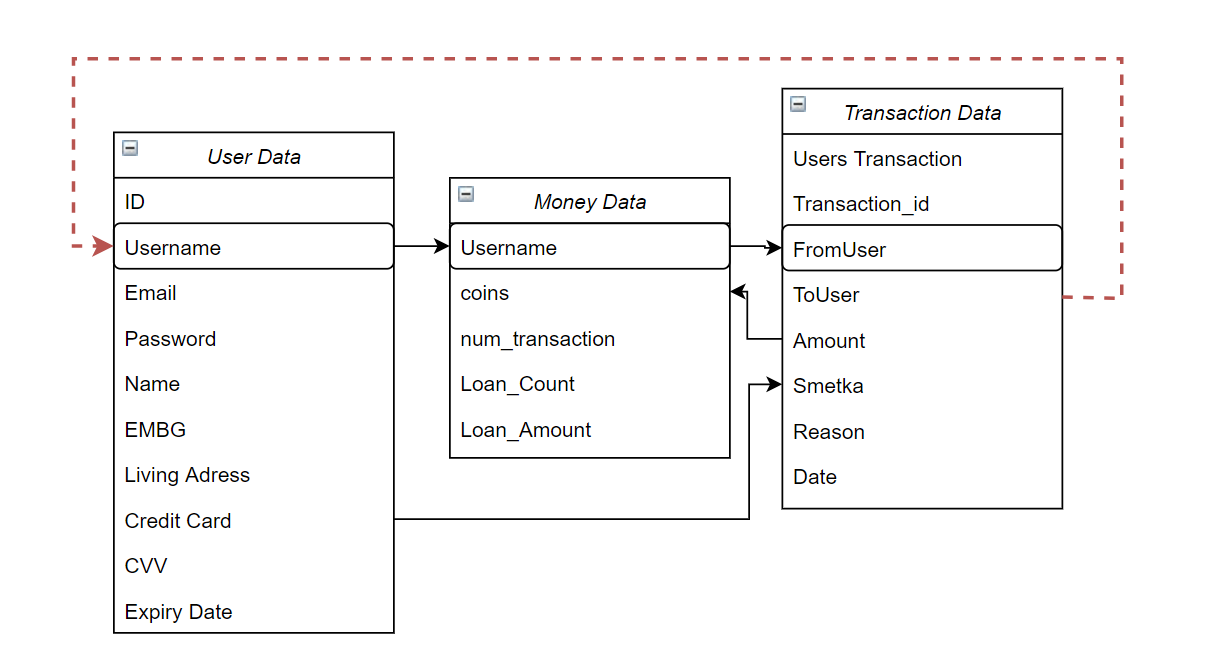


Figure 1.0 - Diagram of all data from the application database

The database for this project uses 3F normalization, which means it uses external keys to link the entity to the spreadsheet and ensure that there is no duplicate data, making it a secure and efficient way to store e-banking data.

Example of how the data structure works: At the moment when the user makes a transaction to another user of the application, first a code analysis is performed (explained further) and if both users are verified, then all the transaction data is entered in the database, but automatically both users are connected to the database, so the administrator and the program can more easily search for them if you need to manipulate transaction history or edit data. Technically speaking, this is done with the help of foreign keys and the use of relational tables.

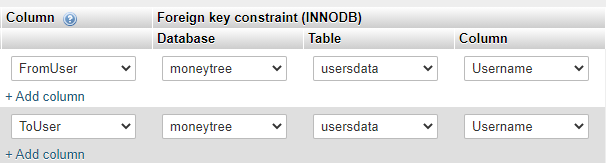


Figure 1.1 - Figure of how the external keys are inserted in the database

# Data security

Security deserves a full page of explanation, because creating an e-banking application should be one of the most secure applications, as it was said, this project was structured around user data security.

One of the ways we provide data is by encrypting everything that the user sends to the server, in this example I will give you my real login information that I use to log in to this web application (MoneyTree)



Figure 2.0 - Image taken from the administrative side of the database, you can see my personal login information on the website

Now that I have given you my email, password and username, you may try to log in to the web application but fail, because the password is encrypted, only the user knows their password, neither you nor I , even though I have all the user information stored on my server, this is because a single line of code encrypts any number of characters into a specific 60-character encrypted code.



Figure 2.1 - Line code that converts the password to an encrypted password

# Using Sessions

A web session is a series of data stored in the browser of the user and the server.

This project uses sessions to track the user while he manipulates his data, we use sessions to track who is logged in and to track how many coins the user has and all his data to enter the platform.

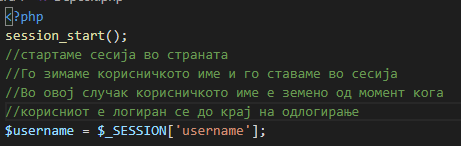


Figure 3.0 - Line code that defines a session in a web page

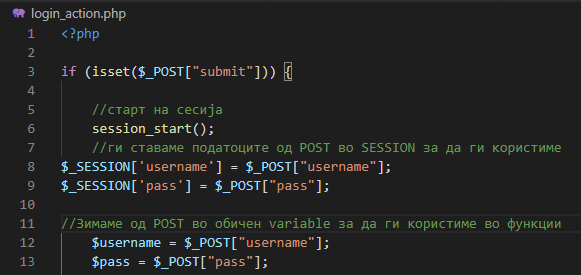


Figure 3.1 - Line code that defines a session in a web page

In the two pictures above we can see session\_start () which is a function that starts the session on the website of the user, and in both pictures we can see that the username has been adopted to be used on the website.

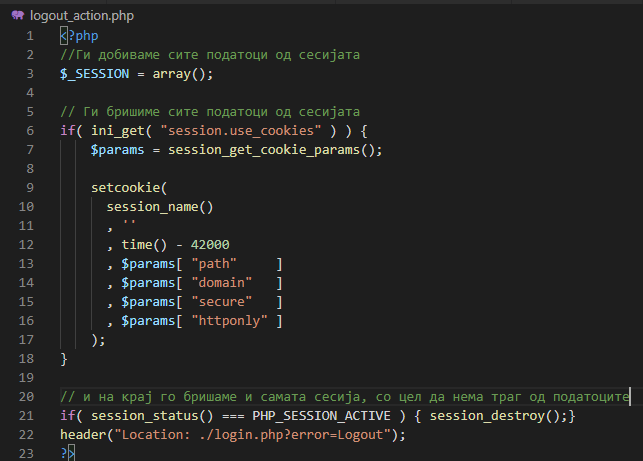


Figure 3.2 - program code for logout

In the same way that we use sessions to log in to a user, we also need a way to unsubscribe from the user and delete the session, we do so through source code which also includes an explanation of the functions and methods used in the comments.

One thing to note is that there is a code "time () - 42000" which means that the session is stored in the browser of users for 4200 seconds or 1 hour, which means that the user can browse the website without should be logged in every second, we give the user 1 hour to use the platform, after that the user logs out automatically, it is used for security reasons, for example, if the user forgot his bank account on the computer, we do not want someone to browse his account, so we automatically delete him.

# Creating an application from a website

As mentioned above, this site uses PWA technology to turn this web page into a web application, this technology is made by Google and requires some standards before making a web page.

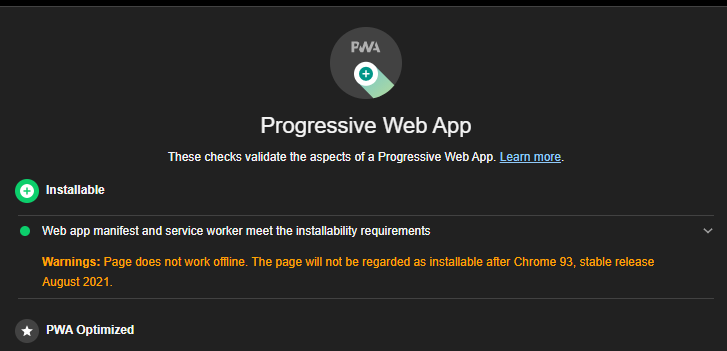


Figure 4.0 - Website analysis to see if it qualifies as a web application

In the image above we can see how the analysis is done using Google Chrome Lightouse, the results that come back say that the website can be installed on any device, it is done by defining a manifest. (Document that defines how the web application will work, the application name, the theme, whether or not there are ads, and the icons used for phones and other small devices.)

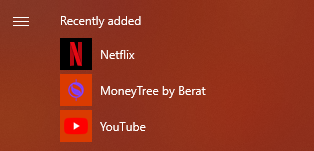


Figure 4.2 - The application installed on the computer



Figure 4.3 - The application installed on a mobile device

A service worker is a type of web service worker (Web Service Worker). It is essentially a JavaScript file that works separately from the browser's main thread, intercepting network requests, caching or restoring cache resources, and delivering push notification messages. This gives the web application the potential to send notifications to the user, to be able to open offline (when the user is not online) but is also essential for turning the website into a web application.



Figure 4.5 - Defining a service worker who will edit a mobile website

The above code instructs the browser to save the cache of all images and the website ready for the user to browse, improves the speed of the website by 300%.

# User interface and design

The design started as one of the first steps to realize this web application, simple user interface attracts every user, especially when the topic is complicated, e-banking websites can be complex, but it is important to be simple for users.



Слика 5.0 - Дизајн Принципи на веб апликацијата

Although there are technologies that facilitate the design by using already made styles, the developer can only change the colors, it is important to create a unique feeling for each web page, so each style code in this web application was handwritten by me .

# Web application hosting

Hosting is the last topic we will look at before we look at how transactions are done and it is important to have the look and feel of the whole web application experience before moving on. But before that, let's explain about hosting ...

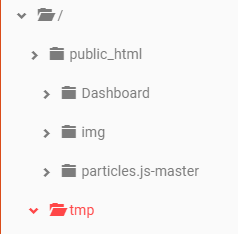


Figure 6.0 - File structure in the 000webhost.com platform

The picture above shows the file system and all its folders hosted on the platform, this is where the website and all its files are stored, but it does not matter, what matters is the red folder written as "tmp" (temporary).

In that folder is stored all the data of the session, as we said above, we use sessions to track the users of the web application, after the user logs out of the web application, the session is deleted and destroyed, but while the user is logged in, sessions are stored on the server, it is the most secure way for users to store their login credentials.

Now that you know the structure of the website and how it is stored on the Internet, you can view the web application via this link - <https://moneytree-berat.000webhostapp.com/.>

# User Login and Registration

When a user registers, the website goes through some features in the code to first check some security checks. Checks if the username has been taken over by another user, whether the e-mail exists in the database, whether the password is strong, and other checks. we know all the information for new users.

# Performing transactions in the platform

Although the website contains many features and many programming codes, we will consider the most unique feature.

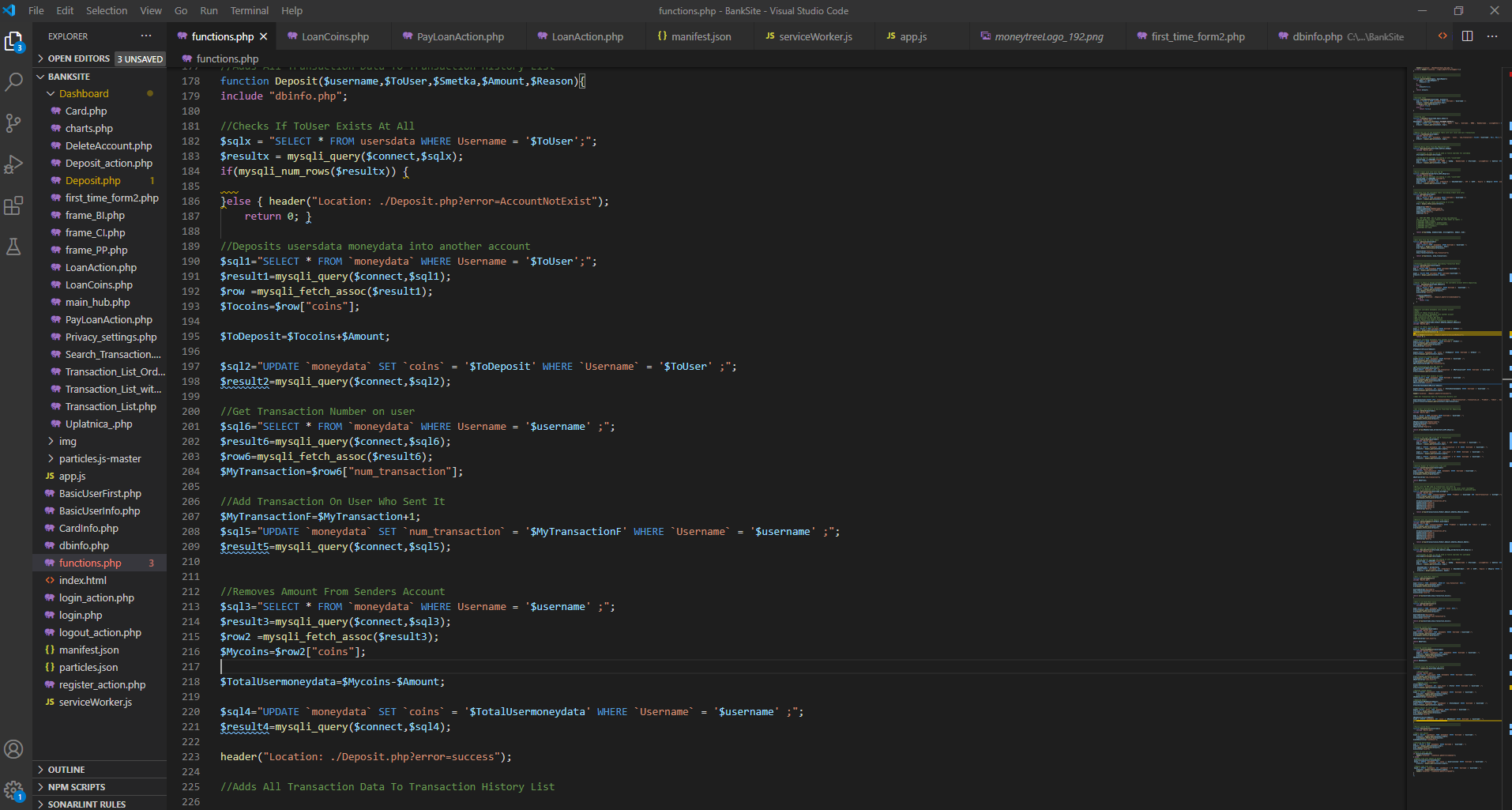


Figure 7.0 - Full code for the transaction function

This feature checks all security checks to check if the user for whom we want to send the money exists or not, if we have enough money to send ...

Once all the security checks have been checked, the feature deposits the money by first taking it out of the user and sending it to the other user, and at the end of all this it will write down all the details of a series of information, details like who to whom sent the money, when and why, in order for the user to have a transaction history or transaction information list.

# Transaction List

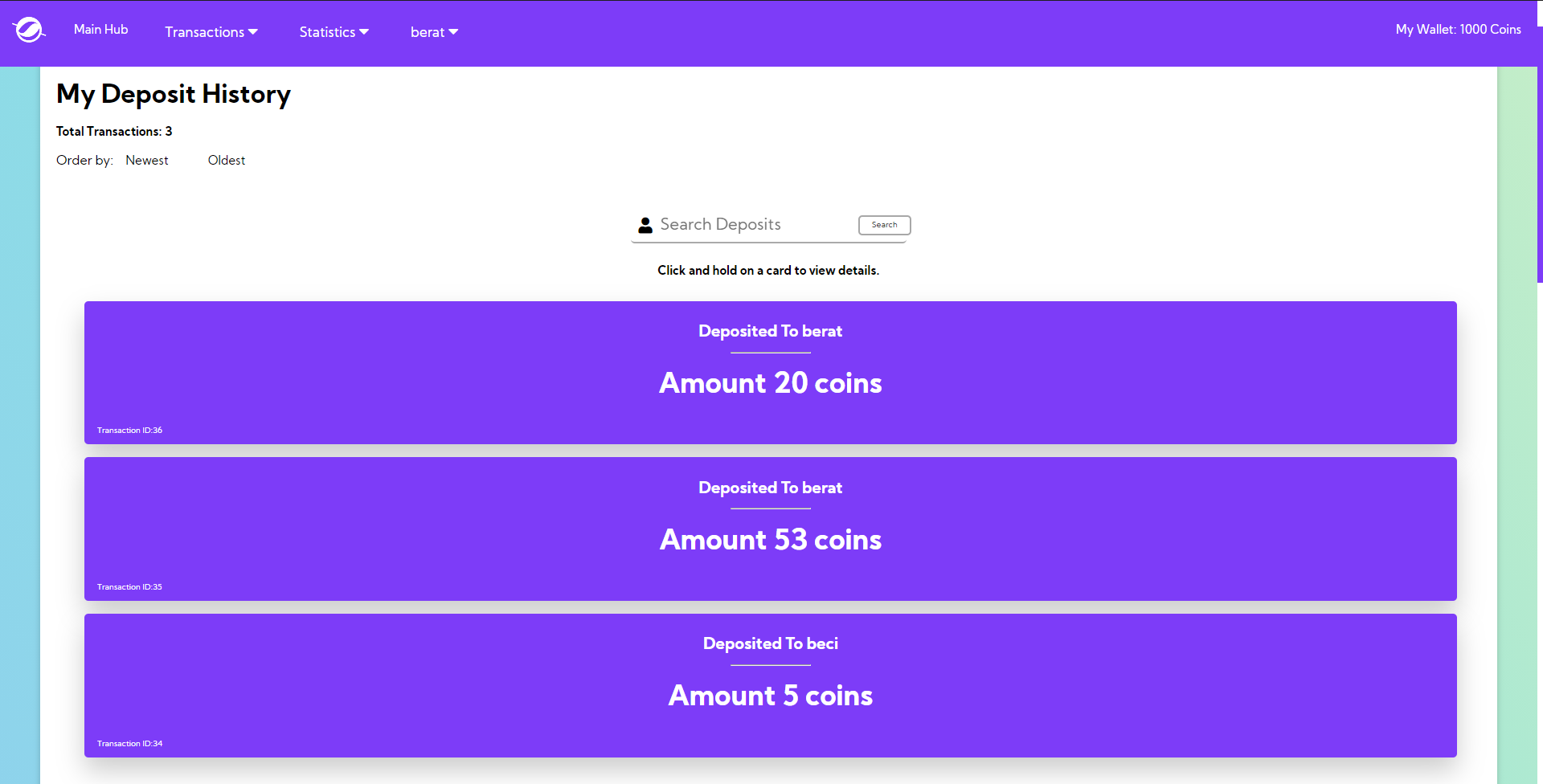
****

Figure 8.1 - Transaction History User Interface

The web application allows users to view all their transactions, such as the application using the data structure we explained on the first pages, it stores the history of transactions in the database and is ready to be given to the user.

The user can manipulate their transactions by selecting a descending or ascending order or even searching for specific transactions. All this is done in a complex way using the links to the database and the "Front-End" Javascript programming language to display all the information accurately and securely.

# Conclusion

This project is a combination of 6,930 lines of handwritten code (yes, I counted them all), 30 pictures, 3 videos and countless hours spent.

Although it sounds a lot, the benefit of knowledge is even more, when I started the project I had no knowledge of linking databases to websites, I did not know what a web application is and I did not know how to program in PHP and use Javascript Libraries.

This document can be huge with information, but this is because it is difficult to put 6,930 lines of code on a 10 page document. But what I hope is that everyone who reads will get a feel for this project.

I will continue to update the website and I will also continue to do other projects with which I will learn even more.

If there's one thing I had to do a lot during the making of this project, it's that I spent a lot of time listening to Kendrick Lamar's "MoneyTrees," which is the song that inspired me to start this whole project.

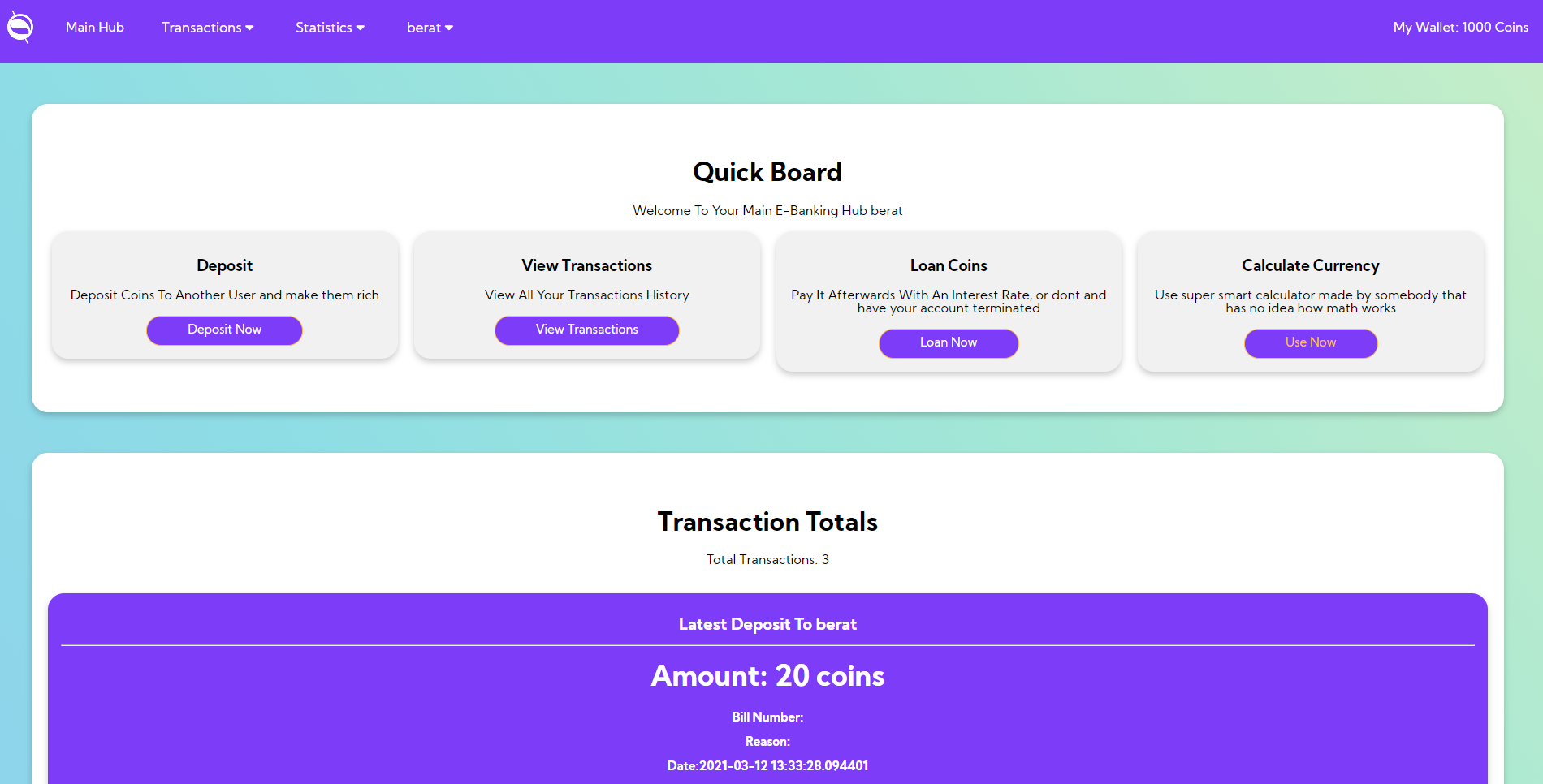


Figure 9.0 - User interface of the web application

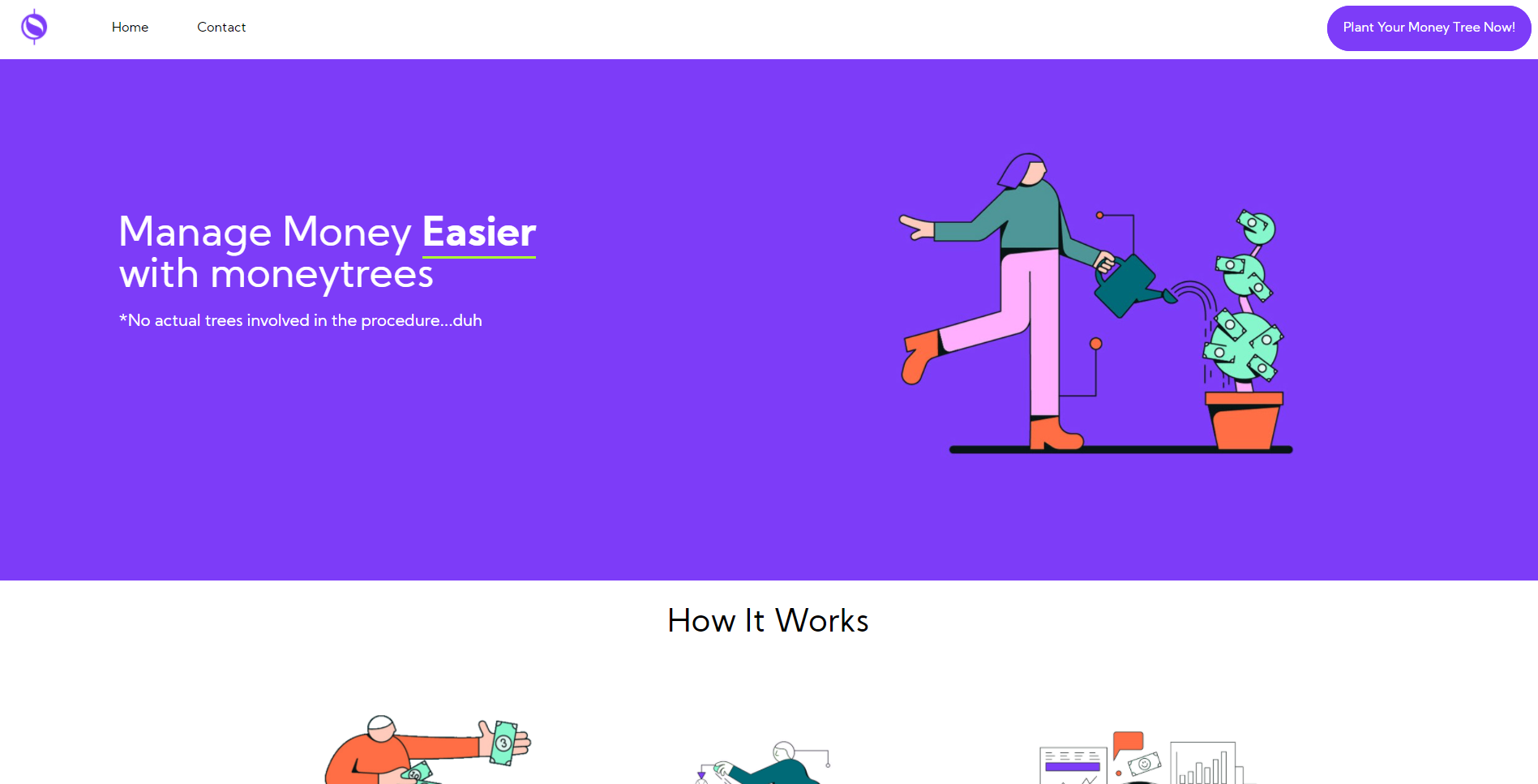
****

Figure 9.1 - User interface of the web application

You can browse the web application via this link: <https://moneytree-berat.000webhostapp.com/>

**Used literature:**

1. [**https://inrekom.org.mk/e-bankarstvo-vo-makedonija/**](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/>