НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

## МАГИСТЪРСКИ ФАКУЛТЕТ

# ДЕПАРТАМЕНТ "ИНФОРМАТИКА"

**ПРОГРАМА „Софтуерни технологии в Интернет”**

**I Курс, 1 семестър**



### КУРСОВА ЗАДАЧА

**КУРС: INFM036 Проект по уебдизайн**

**ТЕМА: Уеб базирано решение за мениджмънт на портфолио от криптовалути**

**НА СТУДЕНТА** : Георги Иванов Георгиев

**Фак. №:** f88683

**Дата:** 12/01/2018 г.

**Преподавател:**

/ Илия Горанов / …………………

София

1. **Цел и задачи на проекта**

Проекта има за цел да създаде базова версия на Single page уеб приложение, за улеснено управление на портфолио от виртуални валути. Основната задача, която седи пред проекта е той да бъде максимално удобен за потребителя и лесен за ползване. Второстепенни цели са висока степен на достъпност (поддръжка на различни, но предимно нови версии на браузъри), както и модерен responsive дизайн, подходящ за всякакъв вид устройства.

1. **Структура, съдържание и workflow**

Проектът се състои от 4 основни страници:

* Начална страница – цели да запознае потребителя със същността на платформата и да изтъкне нейните преимущества
* Register страница – позволява регистрация на нови потребители
* Login страница – позволява аутентикация на потребители със съществуващи акаунти
* Portfolio страница – позволява добавяне на количества от различни по вид криптовалути, както и следене на тяхното развитие (курс и сума на портфолиото)

За пълноценното използване на платформата от потребителя се изисква задължителна регистрация. За успешното регистриране на потребител е необходимо да бъдат въведени следните задължителни данни:

* Име (Name) – пълното име или псевдоним на потребителя. Използва се за обръщение към него в страницата Portfolio;
* Имейл (E-mail) – уникален имейл използван като ключ за ново регистриран потребител. Регистрираните имейли в платформата не могат да се повтарят.
* Парола (Password) – използва се за аутентикация. Полето има следните изисквания – поне 8 символа; поне 1 малка буква; поне 1 главна буква; поне 1 число; поне 1 специален символ
* Повтаряне на паролата (Repeat password) – използва се за удобство на потребителя. Осигурява някакво ниво на сигурност, че потребителят не е сбъркал първоначалното въвеждане на парола, като се изисква от него да го повтори.

След регистриране, потребителят е пренасочен към страницата за аутентикиране. Там той трябва да въведе потребителски имейл и парола, които се сравняват с тези записани към приложението. По този начин удостоверява пред приложението, че именно той е собственикът на данните към даден акаунт. Ако потребителят не въведе правилна комбинация от потребителско име и парола той бива уведомен, че въведените данни са грешни. Ако данните са коректни потребителят се пренасочва към страница Portfolio. Тук той следва да въведе различните позиции(комбинация от вид криптовалута и брой притежавани единици) от своето портфолио. За целта избира бутонът въведи маркиран с иконата плюс. Това действие отваря контекстова страница с лист от криптовалути (комбинация от име и позиция според пазарната капитализация на валутата) попълнен в реално време чрез платформата coinmarketcap и филтър позволяващ филтриране по име. Избирането на желаната валута отваря нова контекстова страница позволяваща въвеждането на размер от желаната валута и прибавянето ѝ към потребителското портфолио. Страницата Portfolio съдържа още две възможности за потребителя – изтриване на цялото портфолио и обновяване на информацията за размерът на портфолиото и цените на отделните криптовалути.

1. **Използвани технологии**

Приложението е от тип Single page application. Потребителят се намира в една единствена уеб страница, а навигирането в нея става посредством подмяна на структурата на DOM дървото. Функционалността на проекта е изградена чрез javascript компоненти. Използваният синтаксис е Ecma Script 6 и HTML 5.

В своята основа приложението разчита на фреймуъркa с отворен код Vue.js. Причината да предпочета използването на този сравнително нов фреймуърк са неговите силни страни, а именно:

* + Малък и олекотен размер
  + Лесното разбиране на същността му и бързото навлизане с цел разработка на приложения
  + Лесната интеграция
  + Добрата документация
  + Възможността му да се комбинира с Ecma Script 6 синтаксис
  + Поддръжката на двупосочна комуникация между компонентите

За реализирането на проекта бе необходимо да използвам следните допълнения към основния фреймуърк:

1. Lodash – библиотека за операции върху масиви
2. Vue-router – библиотека осигуряваща routing в рамките на приложението
3. Vuetify – CSS фреймуърк базиран на т.нар. material дизайн, набиращ все по-голяма популярност. Преимуществата му са минималистичен и функционален responsive дизайн. Фреймуърка използва flex box контейнери за изграждането на рамката на дадена страница.
4. Vue-resource – библиотека за осъществяване на AJAX заявки към API-та на трети страни (в контекста на приложението това е <https://coinmarketcap.com/api/>). Библиотеката обвива функционалностите предоставяни от стандартния XMLHttpRequest.

1. **Програмна логика и връзки между компонентите на уеб приложението**

Предоставеният custom код се състои най-общо от views и components Javascript файлове. View файловете съдържат в себе си голяма част от презентационната и бизнес логика на отделните страници. За разлика от тях component файловете съдържат преизползваема логика и презентация, която е подходяща за повече от едно View или е обща за много на брой View-та. Съществува и трета група javascript файлове – config файлове. Те служат за настройка на средата на Vue приложението.

Съхранението и достъпът до данните е реализиран чрез IIFE конструкции наречени UsersTable и PortfolioTable. Те предоставят на разработчика достъп до „таблиците“ usersTable и portfolioTable, които са част от localStorage.

Prototypes.js е специфичен файл надграждащ базовите обекти предоставяни от javascript с необходима логика за изпълнението на приложението. Стартирането на сайта започва от файл index.html, който съдържа скелета на приложението и главният portfolio-app елемент на приложението, и файл app.js, който създава т.нар. application компонент, в чиито контекст се развиват и разгръщат потребителските действия.

В рамките на приложението са реализирани следните фукнционалности:

* Регистрация и аутентикация – функционалността е реализирана посредством използването на localStorage обект, част от обектите позволяващи записването на локални данни за даден User Agent. Регистрацията попълва usersTable, която е част от localStorage, а аутентикацията сравнява данните въведени по време на Login с тези реално записани в usersTable.
* Портфолио менижмънт – функционалността е реализирана чрез писане по ключ portfolioTable към localStorage. Потребителят записва видове валути и количества към масив от portfolio обекти, съдържащи ключ потребителско име и стойност – самото портфолио на клиента.
* Валидация на данните – функционалността е реализирана чрез стандартни vuetify компоненти, използващи функции за Regular expression, data comparison и др.
* Live data – функционалността е реализирана чрез извършването на Ajax заявки към външно за приложението API, изчакването на резултата от заявката чрез Promise и обработката на данните по подходящ за бизнес логиката на приложението начин.

1. **Изводи и препоръки за развитие на уеб приложението**

Приложението е в начален стадии на развитие. То би могло да претърпи градивни промени по отношение на съхранението на данни (например използването на пълноценна база данни наместо localStorage). Интегрирането на някакъв вид BackEnd сервиращ данни към User interface-а е също добър вариант за разширение. Необходимо е също така да се помисли за хешираща функционалност с цел съхраняване на паролите в нечист вид. Добра идея е да се използва механизъм за embed на външните библиотеки към приложението с цел backup на използваните cdn линкове.

Приложението търпи коментар и разширения по отношение на бизнес логиката.

1. **Описание за необходимите средства за проверка функционалността на приложението**

Приложението може да бъде свалено от гитхъб страница <https://github.com/Georgegig/nbu-altcoin-portfolio/tree/develop> .

След свалянето на приложението се стартира файл index.html. Не е необходимо инсталирането на външни библиотеки и файлове, тъй като те са добавени чрез cdn линкове.