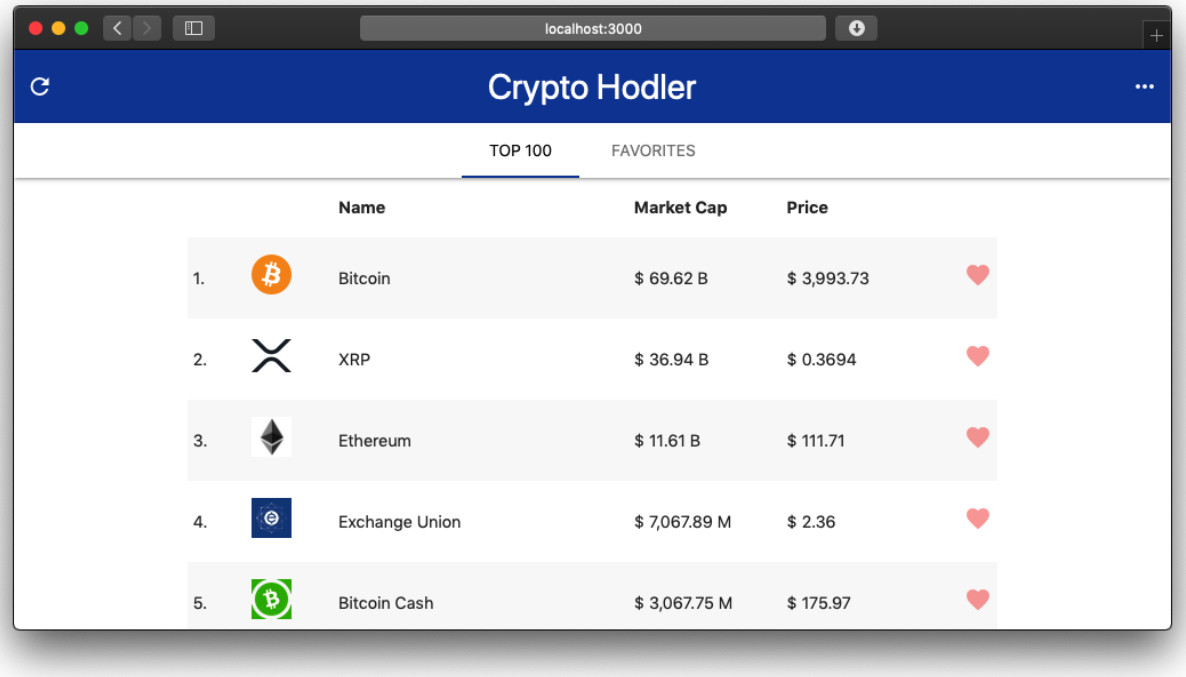
Documentation

Course: www-sovellukset 2018
Assignment: Harjoitustyö
Author: Matvei Tikka, 0453100

About the project

Crypto Hodler is a basic web-app for viewing prices of most popular crypto currencies. User is able to add coins to favorites after signing in. This is how the app looks like:



Requirements

App is very lightweight and universal. It will basically run on almost any modern device that is able to connect to the internet.

User side:

- JavaScript enabled browser
- Internet connection

Server side:

- Nodejs version 10 or higher
- Internet connection

Set-up guide

There are probably multiple different ways in order to get this app up and running. Here are two tested ones:

Cloning repository

In order for this to work you will need:

- Node installed (at least version 10)
- Internet connection

1. Clone repository with a terminal command:

- git clone https://github.com/matikka96/www-harjoitustyo.git

2. After cloning the repository to local device some of the settings must be configured. All can be done in a single file located in *"/config/keys.js"*. Values to be configured will be marked as bold.

```
module.exports = {
        // Create your own Google API keys on Google Developer Console
        google: {
                clientID: 'client-id',
                clientSecret: ' secret-key'
        },
        // Link to database. Create own on mlab.com
        mongodb: {
                uri: 'url-to-mongoDB'
        },
        // Key for encrypting cookies (any string will do)
        session: {
                cookieKey: 'any-string-will-do'
        },
        // API key for CryptoCompare. Create own on their website.
        cryptocompare: {
                api_key: 'api-key'
        }
};
```

3. Execute "server.js" from applications root directory with command:

> node server.js

Now app will be seen here: http://localhost:3000

Running in docker (temporary, API keys are not safe…)

[NOTE] Docker CLI must be already installed and running on local device

Download project from Github and build it in Docker with command:

- *docker build -t hodlerlistserver https://github.com/matikka96/www-harjoitustyo.git*

This will create docker image named "hodlerlistserver".
Final step is to run above image with command:

- docker run -p 3000:3000 -d hodlerlistserver

Now your app is seen here: http://localhost:3000

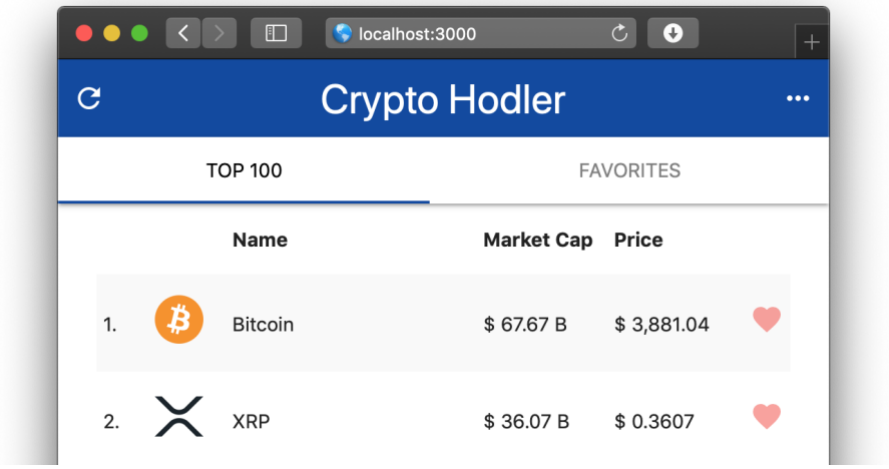
User interface

User interface has been planned with simplicity of use in mind. This is basically a one-page app with few possible action only. All the navigation functionality has been added to the blue bar above.

There are two tabs:

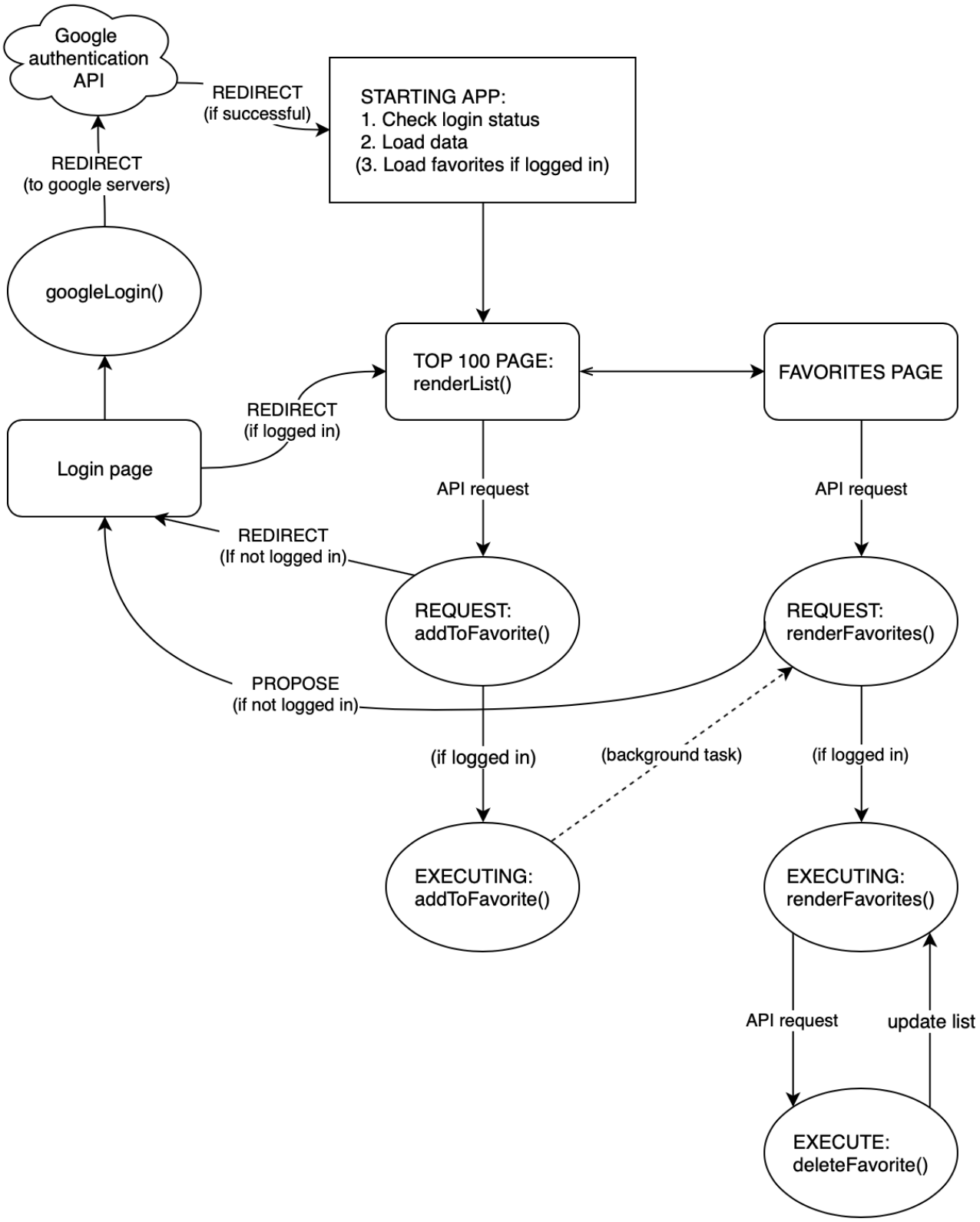
- TOP 100 [List of top 100 coins by market capitalization]
- FAVORITES [Personal favorite coins]

The rest of the screen is reserved for the actual content itself. Heart symbol is a clickable button, that will add current coin to the favorites.



5. Program Design

Program design is well illustrated in the flow-diagram below. Application starts from the square with heading “STARTING APP”.



Technical implementation

Back-end

Heart of the app, which holds whole app together, server so to speak, is running on Nodejs, with the help of package called Expressjs. It for example handles all the routes, renders html files and synchronize with the database

MongoDB is used as a database for this project. Database has been created online using service called mlab.com. It makes whole process really simple and easy to use. Connecting to the database happens with the package called mongoose on file “server.js” on line

App is up and running when the main file named “server.js” is launched. Then it automatically connects to database and is listening for requests from user.

Front-end

Functional part of the app has been implemented with use of Vuejs framework. Most important feature used was definitely reactive list rendering provided by framework.

Visual part on the other hand has been done with help of Materializecss framework. It is responsible for responsive and well-aligned design.

Custom CSS modifications and additions have been made to file named “styles.css” in root directory.

Desired assignment grade

It must be fair to propose the grade based on things that has been done. Let look at the table:

Feature used	**Comment**	**Points**
Responsive	Yes (materializecss)	5
Database	check	5
MongoDB	mongoose	2
Authentication	passportjs	5
Vuejs	check	5
3rd party API	CryptoCompare	3
JSON data transfer	check	3
Dynamic graphics	Materializecss elements	1
AJAX	axios	3
Documentation	check	5
Peer review	(soon) check	5

Total amount of points: 42
Based on the table, proposed grade is **4/5**