FCt FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Interacção Pessoa-Máquina

2020/2021

TMASK

Stage 4: Functional prototype

Realizado por:

41936, Samuel Robalo
44592, Alexander Denisov
50654, Francisco Silva
51095, Daniel Dias

Lab class Nº P2

Group Nº 13

Professor:
Teresa Romão

November 30, 2020

Prototype URL

Online version: https://denisov93.github.io/Tmask/

GitHub Repository: https://github.com/denisov93/Tmask*

Zip file: attached to the e-mail sent.

* email a.denisov@campus.fct.unl.pt for access if required.

Startup instructions

To run the project, it is required to download Visual Studio Code (highly suggested) or a similar tool with npm support.

Then load provided .zip file or use git (https://github.com/denisov93/Tmask) to open project.

The following instructions assume the usage of Visual Studio Code.

To install the npm support for VS Code, open the Extensions menu on the left side bar (ctrl + shift + X) and type "npm". Then install the first two as shown in the figure 1.
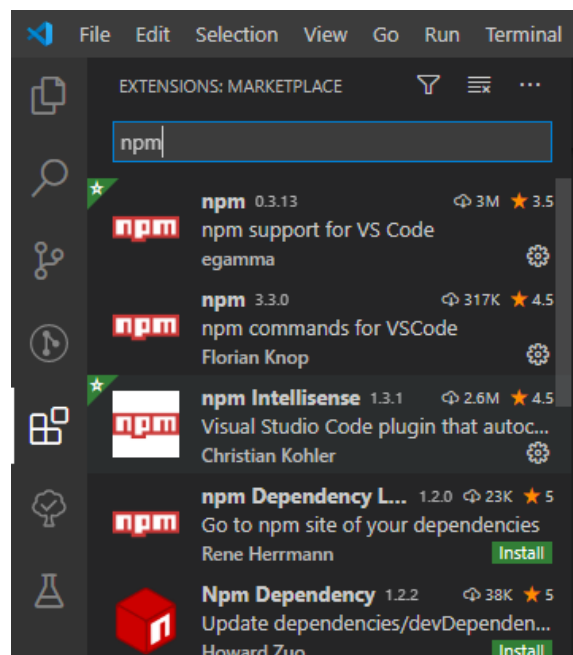

Figure 1: Extensions menu

As figure 2 suggests, to clone the git repository and open project, first open the Source Control menu (ctrl + shift + G) and click on Clone Repository.
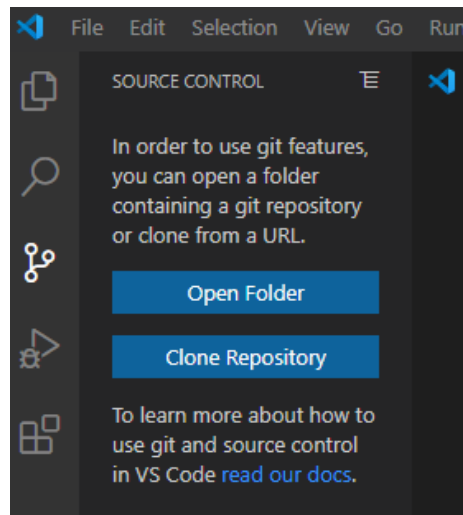

Figure 2: Source control menu

After cloning, the project should soon load, and the Explorer menu (ctrl + shift + E) will look appear similar as the figure 3.
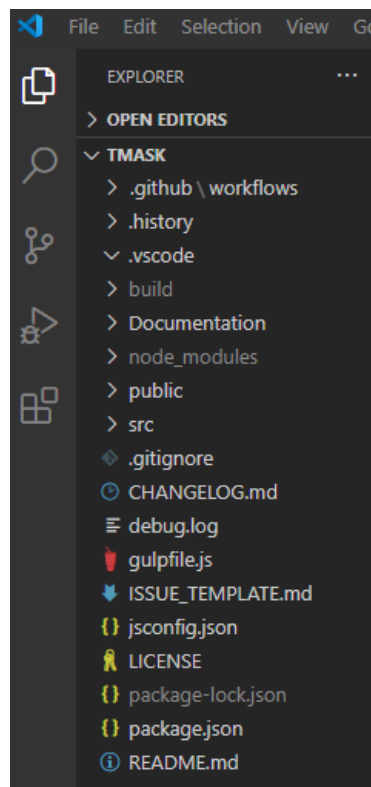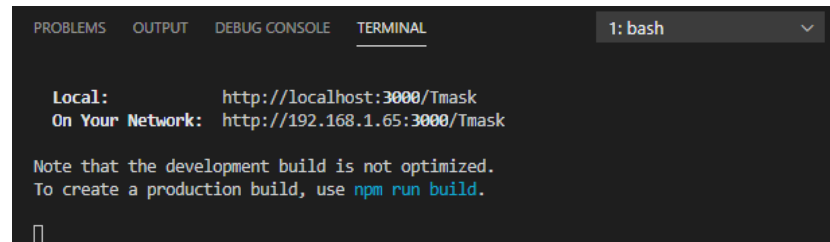

Figure 3: Explorer menu

We now turn on the terminal, if terminal is not visible, the top menu bar View contains the Terminal for display.

This project will set a running server in localhost as the figure 4 shows. For that the localhost port 3000 must not be in use, requires closing any server running or application binding this specific port.

Figure 4: Terminal after a successful build

Firstly, we need to install all project dependencies with command ***npm install***. After this is complete, we need to use ***npm update*** to ensure latest versions of dependencies are up to date, we then launch the server in localhost with ***npm start***. If any issues arrive, repeat the npm ***install*** and ***npm update*** commands and it should be fixed.

A browser window automatically appears after building successfully, with npm start, in the localhost address: http://localhost:3000/Tmask. If that does not happen, the address must be accessed manually.

Briefing

The web application is called TMask, the design being planned and implemented by TacticalDesign (IPM Group 13 of P2 Shift).

The main goal of the project is to create an easy-to-use platform for the average consumer to personalize and purchase hygienical masks. To this end, we will develop a website application that is based on a simple login and shopping flow.

The user creates profiles with their head shape, facial measurements, preferred mask type and number of layers. Then, they can search for masks already created by other users or even use tools provided to make their own mask style and express themselves freely. In that case, they have the option to share their piece with the community.

The idea is that these masks catch the eyes of people walking around or give a feel of a fashionable item.

Scenarios

All scenarios start with a login. The user credentials in the prototype are as follows:

- Username – their name in lowercase.
- Password – any, empty is also accepted.

Scenario #1 – Create a facial profile.

Nahla is using the website for the first time and decides to create a facial profile. She proceeds to take her measurements and introduces the data on the forms.

Nahla has an oval face, 7 cm from the top of the bridge of her nose to just under her chin and 26 cm between her ears, passing through the chin. She picks the "Cloth" mask type as her preference and selects 3 protection layers. Finally, she saves the profile under the name of "Custom profile".

Scenario #2 - Find a matching mask for a dress.

Alicia had a beautiful pink dress with strass to wear to the ball. But, due to COVID, she should always wear a mask when she goes outside.

She looked in her house and found only ugly "medical green" masks.

So, she decided to browse online on a useful website that she knew and searched for a matching mask. She chose the mask named "Simple Pink", selected the facial profile "Alice", and bought it.

Scenario #3 – Share your art.

Pedro, a local artist, was looking for a way to share his art and express himself. But, due to COVID, there is no way to display it in the gallery or in the usual way.

So, to show his paintings, he went to the website that Alicia recommended him. He opened the mask builder, uploaded one of his pictures, adjusted it to its fit and shared them with his audience, filling the required fields. Example for the ones without imagination:

- Title: Art in a Mask
- Description: A mask by Pedro, don't forget to keep in touch with your local artists
- Tags: art painting

To conclude, he opened the catalog and looked for his mask, managing to find it.

Scenario #4 – Jonny spends too much on masks.

Jonny is very afraid catch COVID, so he always uses a mask!

Luckily for Jonny, he found a nice website with masks in different shapes and colors. He chose some masks (e.g.: "The Mask", "Fire" and "The Whale") and even designed one himself (a mask with a green Christmas Tree in the center). He put all of them on the shopping list, but later found out he was about to buy too much. So, he checked what he was buying, removing all but the one he designed.

Project

Incomplete parts

Most of the prototype works as intended for all scenarios, but some features that are not described or needed in the scenarios (such as some buttons) are not completed implemented. There may also be some bugs escaped us.

Tools used

The Web Application was made with **React** and running on a **Node.js server**, we used a **React Template (Argon Design)** as a base to start developing the functionalities of the paper prototype we designed in an earlier phase. We added extra Libraries such as **Font Awesome** that provides icons, **Bootstrap 4** to speed up component development and website responsiveness, the react framework **MaterialUI** for more component functionality and **Konva Framework** where the Mask Editor was built upon (similar to HTML5 canvas). We used **GitHub** for version control and **Visual Studio Code** for the actual development. To publish the Web Application, we deployed the project to **GitHub Pages**.