

Département Mathématiques et Informatique

Filière :

« Génie du Logiciel et des Systèmes Informatiques Distribués »

Module :

« Développement Multimédia »

Projet Multimédia

Editeur de photos

Réalisé par :

Imad TAGHZOUT

Année Universitaire : 2018-2019

1 – Présentation

Nous nous intéressons au développement d'un module permettant de capturer la photo de l'utilisateur pour pouvoir l'ajouter à son profile. Ce module sera utilisable dans les applications Web, Mobiles hybrides et Desktop basées sur les technologies WEB.

Le module sera développé en utilisant une approche incrémentale en trois versions successives :

Version 1 :

Cette version doit comporter :

1. Etape 1 : Une interface pour les actions de la capture
2. Etape 2 : Enregistrement de l'image encodée

Version 2 :

Cette version va améliorer l'interface de la version 1 en ajoutant :

1. Etape 1 : Ajouter à l'interface des composants permettant d'ajuster la qualité de l'image en ajoutant des effets à l'image (contraste, luminosité, teinte, etc...)

Version 3 :

Dans cette version, nous permettrons aux utilisateur de recadrer et de pivoter l'image (cropping). Pour cela, l'interface doit présenter un rectangle (Outline) avec des poignets permettant de guider l'utilisateur dans ses actions.

1. Etape 1 : Tracer l'Outline avec ses poignets de recadrage
2. Etape 2 : Rendre l'Outline et ses poignets sensibles aux mouvements de la souris (Evénements)
3. Etape 3 : Gérer le recadrage (en largeur et en hauteur) et le pivotement selon les événements de la souris
4. Etape 4 : Enregistrement de l'image recadrer

2 – Description des versions

2 – 1 – Version 1

1. L'utilisateur appuie sur le bouton « photo »
2. On affiche l'interface qui contient une « vidéo » et le bouton « capturer ». La vidéo affiche la sortie de la caméra de l'utilisateur. Si le dispositif de l'utilisateur n'a pas de caméra, on affiche et on démarre une petite video en boucle.
3. Quand l'utilisateur appuie sur le bouton « capturer », on affiche un compte à rebour (Count down) de 3 secondes au terme duquel on capture l'image. (On l'affiche)
4. L'image est ensuite encodée sous format Base64

2 – 2 – Version 2

Dans cette version l'interface contiendra en plus des éléments décrits dans la version 1, un ensemble de boutons permettant d'ajuster les réglages de l'image capturée. Un bouton « enregistrer » permettra l'enregistrement de l'image finalisée.

2 – 3 – Version 3

Ajouter un dispositif de recadrage de l'image capturée :

Le rectangle de recadrage doit permettre comporte 8 poignets nommées en utilisant les quatre cardinalités : N pour Nord, E pour Est, O pour Ouest et S pour Sud. Les quatre poignets sont donc : NN, NE, NO, EE, OO, SE, SS et SO :

- Les poignets EE et OO sont sensibles uniquement au recadrage en largeur,
- Les poignets NN et SS sont sensibles uniquement au recadrage en hauteur,
- Les poignets NE, NO, SE et SO sont sensibles uniquement au recadrage en hauteur et en largeur de manière proportionnelle. Une bande en L extérieure à chacune des poignets NE, NO, SE et SO non affichée dans l'image, permettra de pivoter l'Outline de recadrage.

Quand le recadrage est effectué, l'utilisateur appuie sur le bouton « enregistrer » et obtient l'image recadrée.

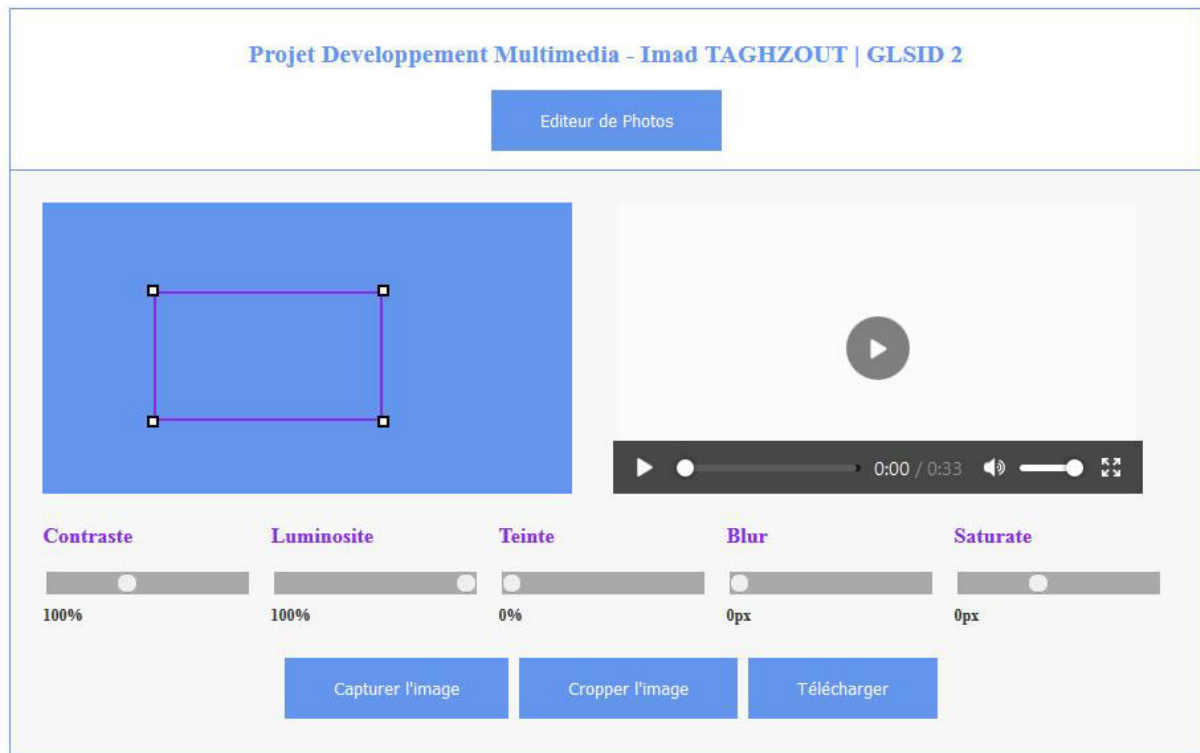
2 – 4 – Version 4

Ça sera la version 3, mais hébergée dans npmjs.org de façon à ce que tout le monde pourra l'utiliser. La même version sera déposée au terme du projet dans un dépôt Github et comportera sa description et les modes de son utilisation dans un fichier « readme.md ».

2 – Capture d'écrans



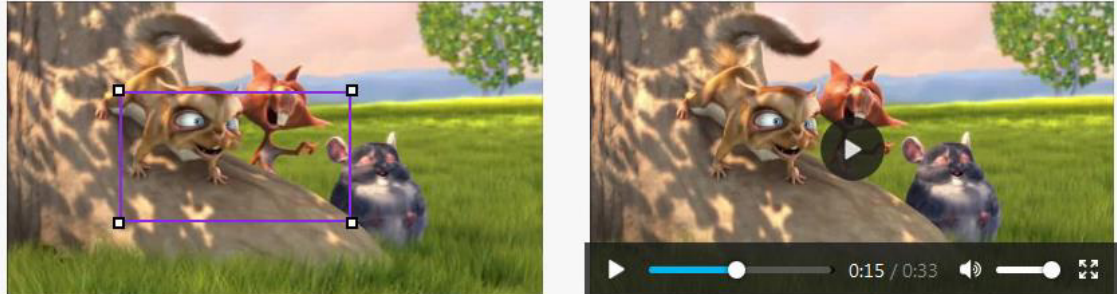
Page d'accueil



Lorsque on clique sur la button "Editeur de Photos", une div qui s'affiche consient les differents fonctionnalités de l'editeur (Effets, Capture, Crop, Téléchargement).

La fonction showeditor () : Permet d'afficher le div qui contient l'éditeur de photos

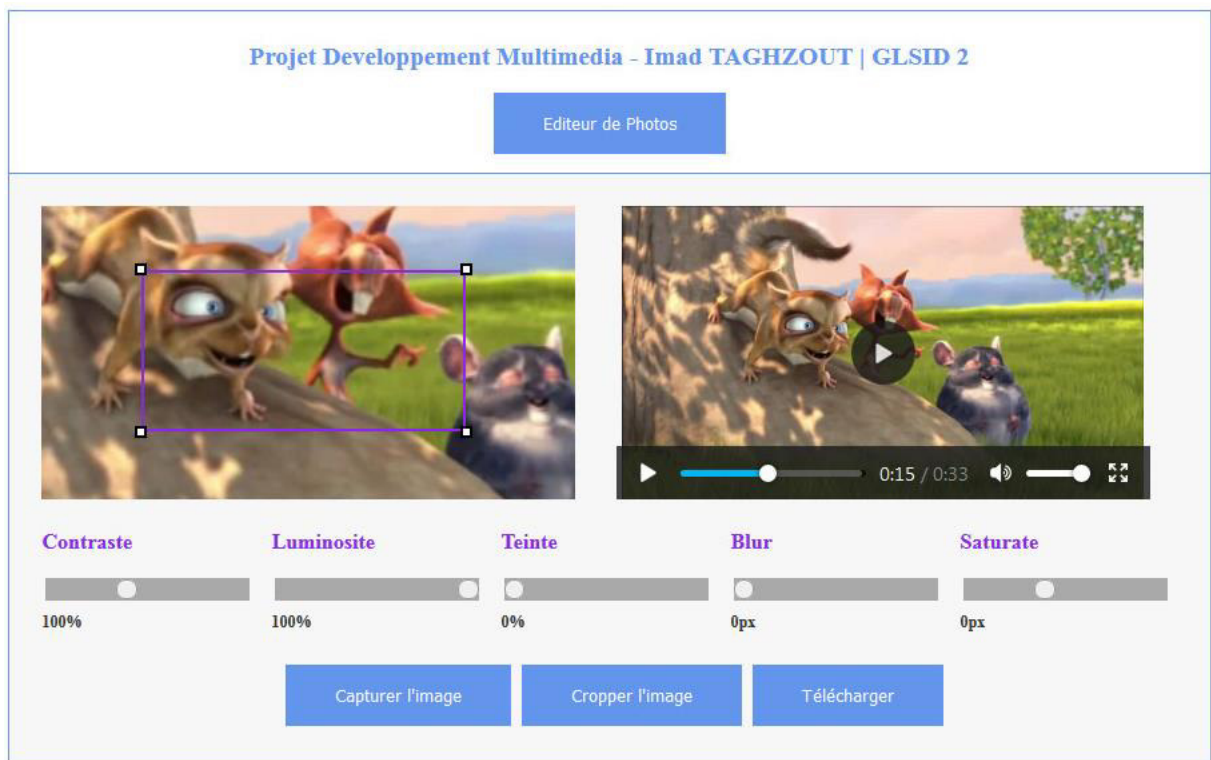
```
function showeditor(){
    document.getElementById("content_editor").style.display = "block";
    document.getElementById("resizable").style.display = "block";
    objeditor.init();
}
```



Clique sur "Capturer l'image", le canvas a gauche va contenir l'image de la video

La fonction capture()

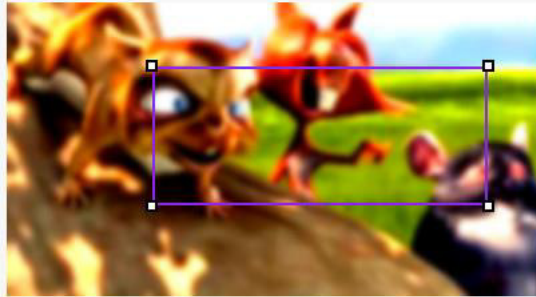
```
function capture(){
    var seconds = 4;
    var countdown = setInterval(function()
    {
        seconds--;
        document.getElementById("btncapture").textContent = seconds;
        if (seconds <= 0)
        {
            clearInterval(countdown);
            objeditor.copy();
            document.getElementById("btncapture").textContent = "Capturer l'image";
        }
    }, 1000);
}
```



Le Clique sur "Cropper l'image", permet de crer une nouvelle image avec des nouvelles dimensions

La fonction cropimage()

```
function cropimage(){
    var destW = 430; var destH = 220; var destX = 0; var destY = 0;
    var sourceCanX = getOffset(document.getElementById('canvas')).left;
    var sourceCanY = getOffset(document.getElementById('canvas')).top;
    var sourceX = getOffset(document.getElementById('resizable')).left;
    var sourceY = getOffset(document.getElementById('resizable')).top;
    var sourceW = document.getElementById('resizable').offsetWidth;
    var sourceH = document.getElementById('resizable').offsetHeight;
    objeditor.ctx.clearRect(0, 0, objeditor.canvas.width, objeditor.canvas.height);
    objeditor.ctx.drawImage(objeditor.img, sourceX-sourceCanX, sourceY-sourceCanY, sourceW, sourceH, destX,
destY, destW, destH);
    objeditor.img = document.createElement("img");
    objeditor.img.src = objeditor.canvas.toDataURL('image/jpeg', 1.0);
    objeditor.canvas.prepend(objeditor.img);
    objeditor.imgFilter = objeditor.img;
}
```



Contraste

210%

Luminosite

100%

Teinte

0%

Blur

2px

Saturate

119%

Capturer l'image

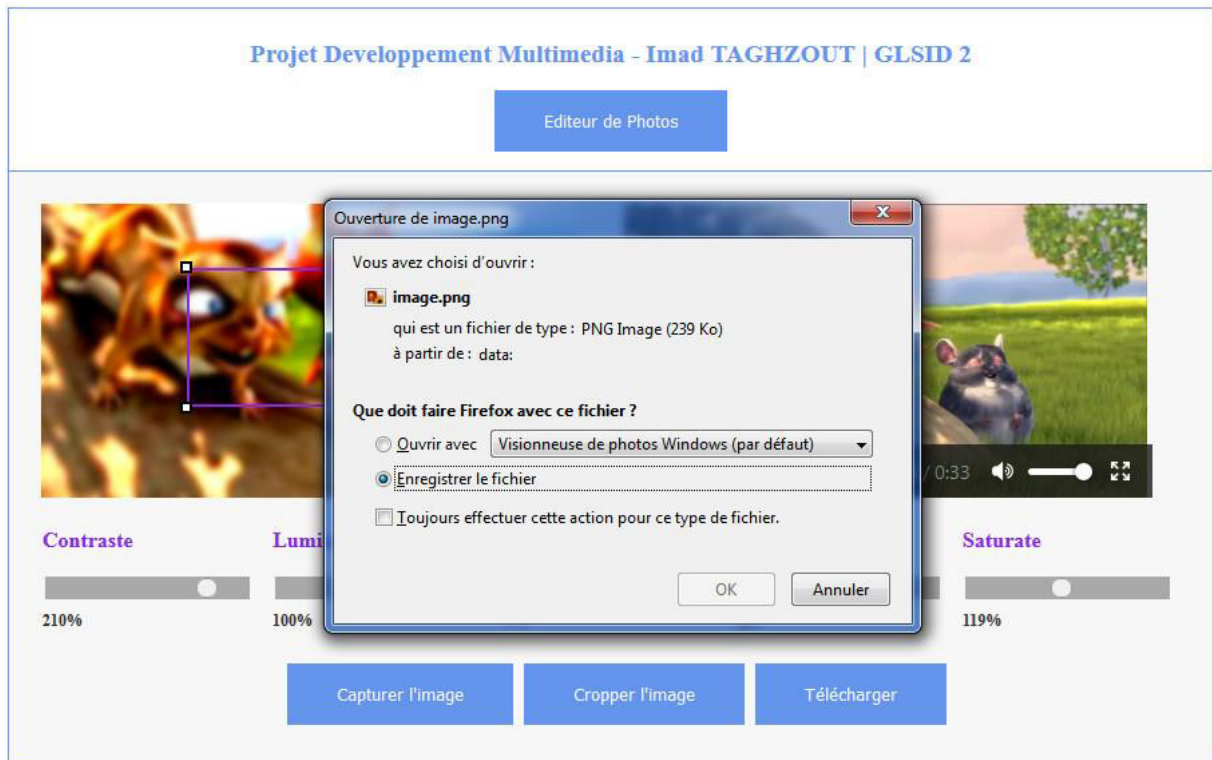
Cropper l'image

Télécharger

Effectuer des filtres sur le contexte du canvas pour donner a l'image des effets

La fonction filters()

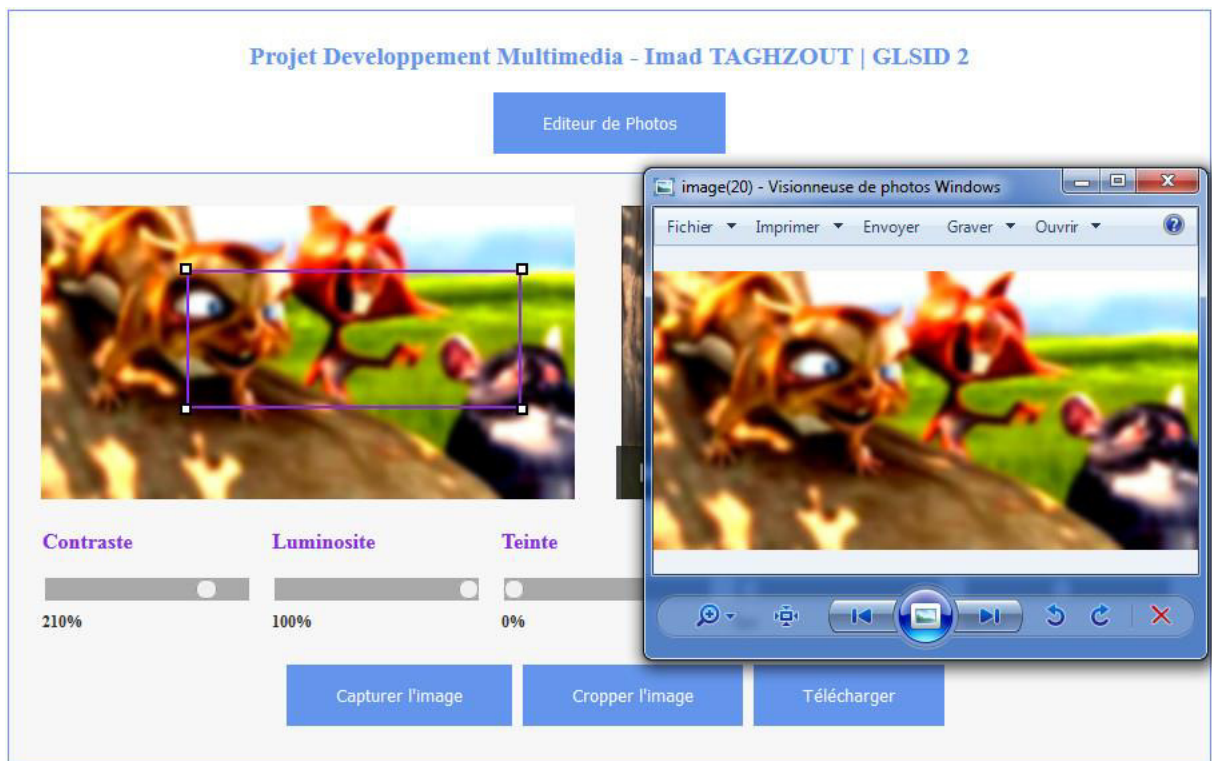
```
function filters(){
    var v_c = document.getElementById("v_c").value;
    var v_b = document.getElementById("v_b").value;
    var v_g = document.getElementById("v_g").value;
    var v_bl = document.getElementById("v_bl").value;
    var v_sa = document.getElementById("v_sa").value;
    document.getElementById("contraste_v").innerHTML = v_c + "%";
    document.getElementById("luminosite_v").innerHTML = v_b + "%";
    document.getElementById("teinte_v").innerHTML = v_g + "%";
    document.getElementById("blur_v").innerHTML = v_bl + "px";
    document.getElementById("saturate_v").innerHTML = v_sa + "%";
    objeditor.ctx.filter = "contrast("+v_c+"%) brightness("+v_b+"%) grayscale("+v_g+"%) blur("+v_bl+"px)
    saturate("+v_sa+"%)";
    objeditor.ctx.drawImage(objeditor.img, 0, 0, objeditor.w, objeditor.h);
}
```



Le Clique sur "Télécharger", permet de télécharger l'image apres l'application des filters

La fonction download()

```
function download(){
    var download = document.getElementById("download");
    var image = document.getElementById("canvas").toDataURL("image/png").replace("image/png", "image/octet-stream");
    download.setAttribute("href", image);
}
```

Voila l'image après le téléchargement

3 – Déploiement en module 'taghzouteditor' avec npm

```
C:\>cd C:\Users\imad\Documents\GitHub\projet-multimedia
C:\Users\imad\Documents\GitHub\projet-multimedia>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See 'npm help json' for definitive documentation on these fields
and exactly what they do.

Use 'npm install <pkg>' afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (projet-multimedia) taghzouteditor
version: (1.0.0) 1.0.0
description: Module editeur photos filters
entry point: (script.js) script.js
test command:
git repository: (https://github.com/usqlar/projet-multimedia.git)
keywords:
author: imad taghzout
license: (ISC)
About to write to C:\Users\imad\Documents\GitHub\projet-multimedia\package.json:
{
  "name": "taghzouteditor",
  "version": "1.0.0",
  "description": "Module editeur photos filters",
  "main": "script.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/usqlar/projet-multimedia.git"
  },
  "author": "imad taghzout",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/usqlar/projet-multimedia/issues"
  },
  "homepage": "https://github.com/usqlar/projet-multimedia#readme"
}

Is this OK? (yes) yes
C:\Users\imad\Documents\GitHub\projet-multimedia>
```

La bonne chose à propos de la publication d'un package NPM est la connaissance minimale requise. après l'exécution de la commande suivante: **npm init**

on a obtenir le fichier **package.json** qui contient les informations de module (nom de package, auteur, etc...)

```

C:\Users\imad\Documents\GitHub\projet-multimedia>
C:\Users\imad\Documents\GitHub\projet-multimedia>npm publish --access=public
npm notice
npm notice package: @imadtag/taghzouteditor@1.0.0
npm notice === Tarball Contents ===
npm notice 518B package.json
npm notice 66B .gitattributes
npm notice 4.1MB bunny.mp4
npm notice 5.8kB index.html
npm notice 3.7kB script.js
npm notice 2.8kB css/design.css
npm notice === Tarball Details ===
npm notice name: @imadtag/taghzouteditor
npm notice version: 1.0.0
npm notice package size: 4.0 MB
npm notice unpacked size: 4.1 MB
npm notice shasum: 0612d9c69c9cee7d0ffc908d54323ce073c5dac7
npm notice integrity: sha512-3Pk1XCTMXgXmI [... ]UBx/QtXL9UjNw==
npm notice total files: 6
npm notice
+ @imadtag/taghzouteditor@1.0.0
C:\Users\imad\Documents\GitHub\projet-multimedia>

```

Ensuite on va publier notre module avec la commande **npm publish --access=public**

The screenshot shows the npm website interface. At the top, there's a navigation bar with links like 'npm Enterprise', 'Products', 'Solutions', 'Resources', 'Docs', and 'Support'. Below this is a search bar and a user profile section for 'imadtag'. The profile section has tabs for 'Packages', 'Profile', 'Tokens', and 'Billing'. Under the 'Packages' tab, it lists '3 packages'. The first package is '@imadtag/photoeditor', the second is '@imadtag/tageditor', and the third, which is highlighted with a blue box, is '@imadtag/taghzouteditor'. The description for '@imadtag/taghzouteditor' is 'Module editeur photos filters'.

Voila le module 'taghzouteditor' qui a été créer
(une capture des packages sur mon compte npm).

@imadtag/taghzouteditor

1.0.0 • Public • Published a minute ago

Readme

Admin

0 Dependencies

0 Dependents

1 Versions

Unable to find a readme for @imadtag/taghzouteditor@1.0.0

Keywords

none

install

> npm i @imadtag/taghzouteditor

version	license
1.0.0	ISC
homepage	repository
github.com	github

Le package avec la commande d'installation npm i @imadtag/taghzouteditor

The screenshot shows a Visual Studio Code window with a project named 'testnpmmodulepack'. The Explorer sidebar shows the file structure: 'index.html', 'package.json', 'package-lock.json', 'node_modules', '@imadtag', 'taghzouteditor', 'css', 'design.css', '.gitattributes', and 'bunny.mp4'. The main editor shows the 'index.html' file with the following content:

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <title> Projet Developpement Multimedia - Editeur de photos </title>
6   <script src="script.js"></script>
7   <link rel="stylesheet" href="css/design.css">
8 </head>
9 <body>
10
11   <!-- cadre de cropped -->
12   <div class='resizable' id="resizable">

```

The Terminal panel shows the output of the command 'npm i @imadtag/taghzouteditor':

```

Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\testnpmmodulepack>npm i @imadtag/taghzouteditor
npm WARN saveError ENOENT: no such file or directory, open 'C:\testnpmmodulepack\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'C:\testnpmmodulepack\package.json'
npm WARN testnpmmodulepack No description
npm WARN testnpmmodulepack No repository field.
npm WARN testnpmmodulepack No README data
npm WARN testnpmmodulepack No license field.

+ @imadtag/taghzouteditor@1.0.0
added 1 package from 1 contributor and audited 1 package in 8.733s
found 0 vulnerabilities

C:\testnpmmodulepack>

```

Création d'un projet test, pour tester l'installation de module "taghzouteditor"

Conclusion

Ce projet de développement multimédia consiste a créer un éditeur de photos a savoir l'implémentations des différents fonctionnalités (filtres, crope, capture, etc..) , a été très bénéfice pour moi au niveau d'amélioration des techniques et des méthodes de développement web.