

# Project: Image Gallery

jan.schulz@devugees.org

# Preparations

- 1 – Pull/Clone the latest [github.com/foobaroo/devugees8](https://github.com/foobaroo/devugees8) repository.
- 2 – There, copy the folder `/javascript/projects/imagegallery` into your local Desktop/imagegallery (for example)
- 3 – step into your local Desktop/imagegallery and run **npm install**
- 4 – after the node moduls have been installed, run **node server.js**
- 5 – Go to your browser and open `http://localhost:3000` , there you should see an empty website
- 6 – Therefore, go into the `public/` folder and take a look at the `index.html` and `styles.css` – These are the files that you need to work with

# Task 1

Implement the responsive design as visualized in the following slides with example images of your choice. If you want, you can use Bootstrap.

- Each image is 350x200 pixels in thumbnail mode
- When the user clicks on a thumbnail, the detail mode opens
  - A semi-transparent div will be shown with height and width 100%
  - On top of that, one div containing multiple divs showing the picture and „next“ and „previous“ buttons or arrow images
  - When the user clicks on the transparent div or on the cross, the detail mode will be left
  - In the mobile version, there is no transparent div needed

# ImageGallery

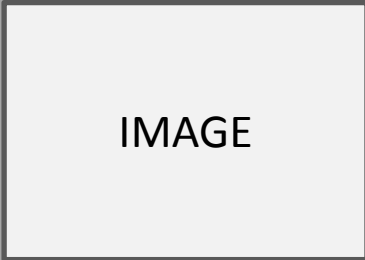
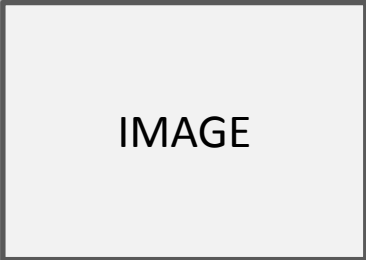
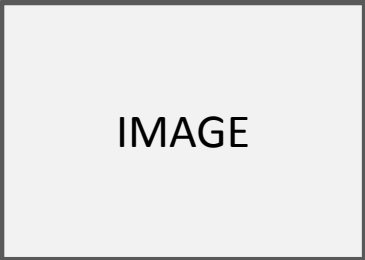
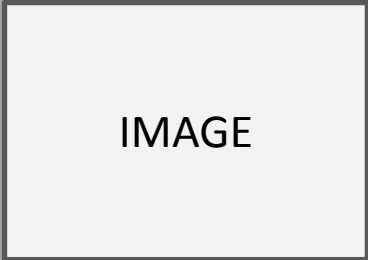
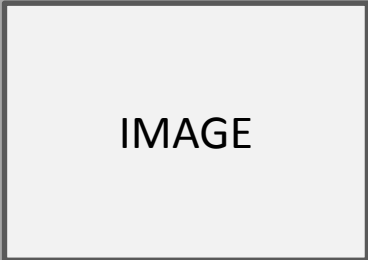
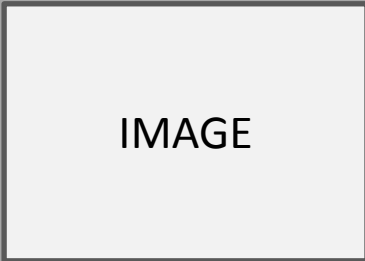
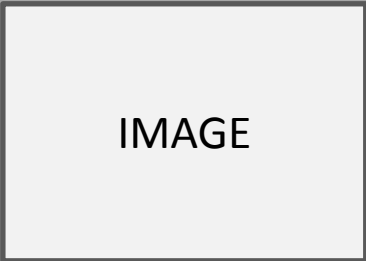
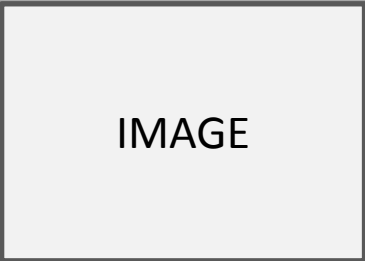
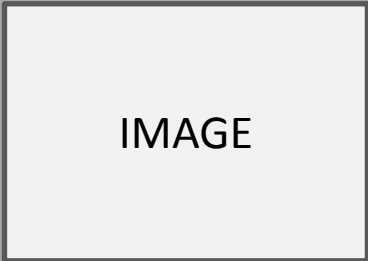


Image Detail: Desktop



## Upload Image: Desktop

### ImageGallery

Take a look at  
Fileuploadexample.html  
For an example form

Upload Image: Desktop

Image Gallery

5

After the Image Upload was successfull,  
The server will redirect you to  
<http://localhost:3000/>

ImageGallery

Search ...

GO

UPLOAD IMAGE

IMAGE

IMAGE

IMAGE

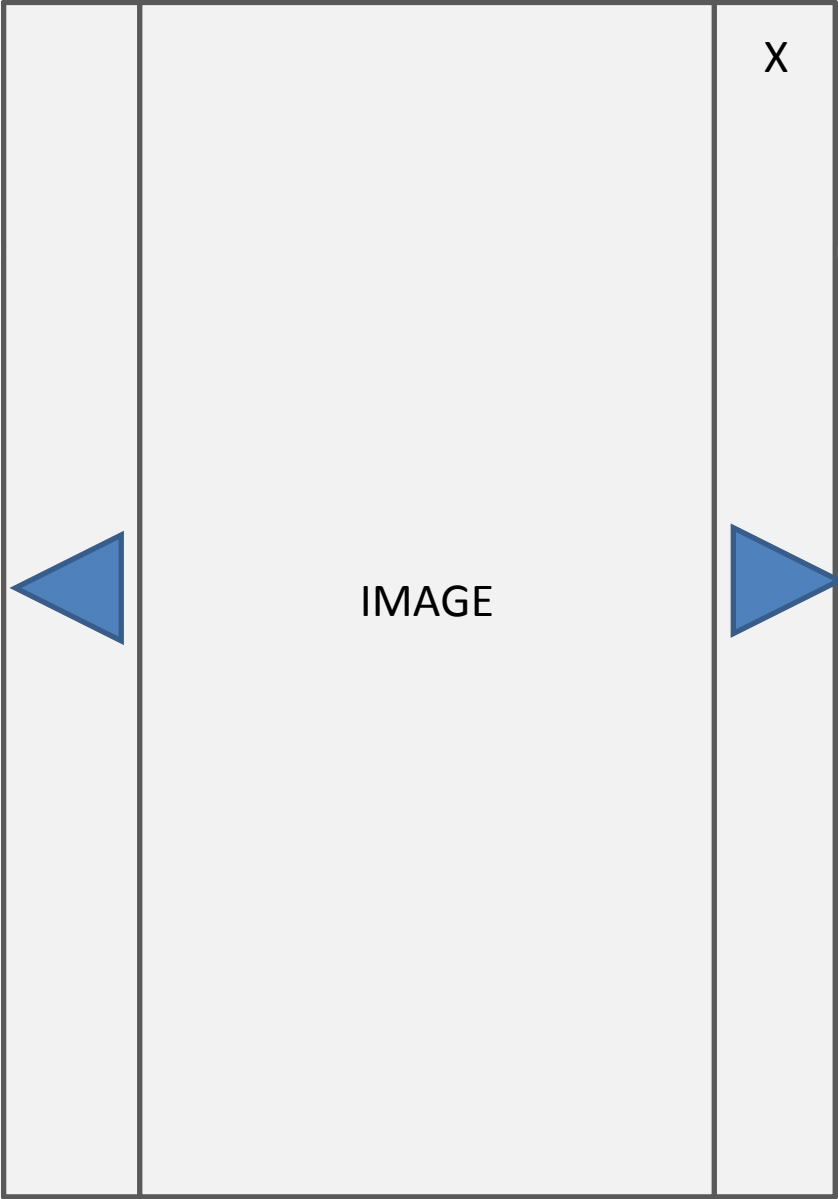
IMAGE

IMAGE

IMAGE



Starting Page: Mobile



Upload Image: Desktop

# ImageGallery

Search ...

GO

UPLOAD IMAGE

Select File ...

BROWSE

Tag1, Tag2, Tag3

Upload Image

# Task 2

Test the following API endpoints with Postman or your browser, except c) which you should test with fileuploadexample.html

- a) GET <http://localhost:3000/search> - returns all images
- b) GET <http://localhost:3000/search?q=tiger> - returns all images that have at least one tag that includes the string „tiger“
- c) POST <http://localhost:3000/upload> - uploads a new image

1 - Analyze what you get as response.

2 – Extend your application to

- load all picture urls as thumbnails and details using the API endpoint a)
- load certain picture thumbnails using the search textbox using b)
- upload an image using c), take a look at the example upload form in fileuploadexample.html and try to use that code for your own webapp

# Hints

- Take a look at
  - `Fileuploadexample.html` – Example how to upload a file to the server including tags etc.
  - `Stoppropagation.html` – about the propagation of events and how to control it
  - `Delegateexample.html` – delegation of click events for dynamic added elements to the DOM tree