

## Project description

**Clarification:** There are four different tasks during this lecture.

1. During the tutorial there are voluntary **Breakout sessions** and **Round up Quizzes**. These are for your own benefit and you are very welcome to complete these at any point in time.

The solutions for these are discussed during the Q&A sessions for each exercise sheet.

2. On each exercise sheet, there are mandatory **individual tasks**. These have to be handed in over Uni2Work by each student individually. Solving these Exercises is required to pass the lecture.

While Questions can be discussed with other students, please keep those discussions to a broad level and submit individual solutions. There will be no direct solutions discussed during the Q&A Sessions, but feel free to ask questions.

3. On each exercise sheet, there are voluntary **group tasks**. Here, the solution for the task can usually be directly used for the final project. Therefore, we highly recommend doing the tasks during the semester. Further, it is possible to hand in your solution to receive valuable feedback. There will be no direct solutions discussed during the Q&A Sessions, but feel free to ask questions. The group (**3-4 people**) for these exercises should be the same group you submit your final project with.

**Plagiarism:** Unless otherwise specified, it is fine to use libraries or code snippets from the internet. In case a third party library is used, a source, (short) explanation and the corresponding license should be noted in the code. For code snippets the source has to be clarified. Further, the function of the code has to be explained (for longer code snippets explain every main section).

For individual tasks do not copy work from other students and for groups tasks / final project, do not copy code from other groups. In cases where you expect a similar solution, please note that accordingly, e.g. "We were stuck and asked group 2, to explain us feature XX. Our implementation follows their explanation."

In case any plagiarism is found, the corresponding task will be evaluated with 0 points and we will discuss further steps.

## Final Project

Your task is to create a website to generate, download and display memes ([funny] pictures, gifs or videos with text). The site should allow a user to upload (or select) an image, add text and download the created image. This should be possible using either an API call or a GUI. Other users then should be able to view the created images.

In the following, we have compiled a list with features and corresponding points. The features marked in **red** are mandatory to receive a passing grade. **You cannot make up for red points with other features.**

Different features in one box are upgraded versions. If one feature is completed, all above are done as well, e.g. if you provide a WYSIWYG Editor you will not need (additional) text formatting options, however you receive those points as well.

Own features can be added (or listed features modified) and will receive points according to their complexity (and novelty compared to other features). However, we will not be able to specify points for each individual task before the final submission.

In general, the website should be implemented in a coherent way, i.e. the features should be usable and useful. For example, it does not make sense to implement a filter function but it is not possible to apply that filter to the images.

The list below sums to 135 points. Starting from 0 points every 10 points will increase your grade by one step:

Points	Project-Grade
10 (all red points!)	4.0
20	3.7
30	3.3
40	3.0
50	2.7
60	2.3
70	2.0
80	1.7
90	1.3
100	1.0

**Any points above 100 will not improve your grade or provide any further benefit. Therefore, you do not need to implement all features listed.**

**Two third of your final grade will be the project and one third the oral exam.**

**The project is to be handed in around two weeks after the last exercise sheet. The exact date will be announced in time.**

## Feature list

### Template images:

The user shall see all options for starting a meme with a selection of already available images or options to provide a source for an own image.

Feature	Points
Provide images stored locally on the server AND uploaded by the user	1
Provide images downloaded from a user provided URL	1
Provide images downloaded from ImgFlip API	1
Screenshot from user provided URL	1
Hand- (mouse-) drawn	2
Photo from connected camera	1
Display available images / options to the user (slide show, drop down, overview, etc.)	1

### Editing a template:

Here, all options to edit the chosen template should be present. Adding text, generating the final image, adding other images, etc.

Feature	Points
Show the current state of the meme to the user	1
Provide two text inputs for top and bottom and corresponding coordinate fields to adjust the position. Additionally one textbox for an image title	1
Add basic text formatting options for the text inside the image (colour, size, bold, italic)	2
Provide a WYSIWYG editor for all text inside the image	2
Option to add more than two texts to the image	2
Append other images (the options to choose additional images might be limited compared to template images) left/right/bottom/top to the current image	3
Adjust canvas size freely and add images everywhere	4
Next / Previous button to cycle through templates (keep other settings, e.g. already entered text)	1
Full navigation through all template options (drop down, sidebar, popup or others) (keep other settings, e.g. already entered text)	2
Statistics chart (how often are templates viewed/chosen/generated/viewed (after generation), up voted, etc.)	3
One generate button (local generation, server side or third party provider)	1
Two or three different generate buttons (local generation, server side or third party provider)	1
Two or three generate buttons with the option to choose a maximum file-size before generation (which the generated image then does not exceed)	2
Display the generated image in any way	1
Display the generated image in any way and provide a download button next to the image	1
Display the generated image in any way and provide a download button as well as share buttons next to the image	1

### API:

A structured API should give access to some of the sites' functionality without requiring a graphical interface. The API should provide image creation and image retrieval.

Feature	Points
Create a single image with bottom/top text	1
Create a single image with multiple textboxes at chosen locations and text format.	2
Create a set of images (provided as a zip file), e.g. from one image but a list of different texts	3
Get any previously created image	1
Get a set of images (provided as a zip file) using search parameters (see search). Specify a Maximum amount of retrieved images!	2

Viewing generated memes of others:

Everyone should be able to view generated memes. There can also be many additional features like up/down votes, comments, etc.

Feature	Points
Overview: present an overview showing image and title (e.g. endless scroll)	1
additionally provide passive information per image (views/votes/comments)	1
additionally provide interaction for each image (up/down vote, download, share)	3
Single View: present one image with simple passive information in a slide show (including left/right buttons)	1
Present the information in graphical form (up votes/views over time, portion of created images, portion of views, etc.)	2
Additionally, add a random button	1
additionally add an auto play button (goes to next image every X seconds) with the option for going through images random or ordered	2
Add sort/search/filter options to both pages (see search)	2

Search:

For various features, a search function / filter (or sort function) is needed. The following properties could be used.

Feature	Points
Filter / sort after creation date	2
Additionally, filter / sort after views or votes	2
Additionally, filter after file format or used template and search after (parts) of the title or use tags	2

Authentication, Meme history, and personalized functions:

If individual users can be identified, a set of features can be added.

Feature	Points
Any form of authentication / logging in	2
Sign in with eMail/Name and Password	2
On meme creation, option to mark as private/unlisted/public	3
Show own history of created memes	2
Including unfinished drafts (add save as draft and continue draft options)	2
Account bound votes and comments.	5

Additional features:

Feature	Points
Implement a video stream (actual video) showing memes (like auto play in the slide show view)	3
encrypt the Video stream	1
Include gifs as possible templates (static caption)	2
Include gifs as possible templates (changing caption)	2
Include videos as possible templates (no caption, only upload)	1
Include videos as possible templates (static caption)	3
Include videos as possible templates (changing caption)	5
Speech to text: allow for dictating captions	5
Implement a full voice control including dictating text (e.g. selecting template, etc.)	10
Text to speech (Accessibility): Screen reader, reading the title	4
Screen reader, reading title, explaining the image content and reading the captions	5

Implementation Details:

Feature	Points
Testing: Write automated tests	3
Write automated tests covering 100% of the code	3
Everything is implemented in React	2
Code: well-structured	2
well-structured and some comments	2
well-structured and fully documented	2