

PHOTO GRID

- Remember together -

CONCEPT:

PHOTO GRID is a site for people to connect by sharing images. There are a few sites like this already, the big difference is that on most social media platforms you upload images to show other people things they did not participate in. On this site you will only be able to view images if you attended the event, or if someone was kind enough to give you an accesstoken.

The first step is to create an account, after that the user can start to create and connect to grids. When a user creates a grid a string of characters is generated that can be shared with friends that also attended that specific event eg. weddings, after works, holidays, graduations, class reunions. When that is done, everyone connected to the grid can upload images that will be displayed as a Photo Grid.

Another way to use it could be to create a Photo Grid for a newborns first year. Photo Grid could also be used to gather images of a loved one that is no longer alive to make an online memorial page.

THOUGHTS & CONCERNS:

A concern of mine regarding this project is that it feels as if it's getting pretty big. I have tried to grade different parts of the site from 1-3 to see how balanced the tasks are. Overall it seems to be rather balanced. But balanced or not, there is still a lot to do.

The good thing with it being bigger is the fact that this is something that can continue to work with after the bootcamp is over. I want to do something where I have the possibility to learn a lot and I believe I can do so with this project. I also feel as if this is a project that will show potential recruiters what I'm capable of in a straight forward way. There is also the possibility to make it into a react native app in the future which of course would be great fun.

When I'm done I hope that I have:

- Learned how to upload images to a database
- Learned how to retrieve images from a database and displaying them on a page
- The knowledge to generate some sort of accesstoken that only can be used if logged in
- Learned how to delete something from a database with the help of a counter
- Learned how to connect several mongoose models with each other
- Used CSS grid in a more advanced way
- Increased my React and Redux skills
- Advanced my CSS-in-JS skills
- Come to an understanding of what I'm truly capable of
- And a whole lot more!

THE PROJECT DIVIDED UP:

Not signed in:

- **Header [1]**
 - Header title with under title [1]
 - “PHOTO GRID – remember together” [1]
 - Make text a clickable <a> returning to start page [1]
 - Sign up/log in button that links to sign up/log in page. [1]
 - Handle sign up in backend [1]
 - Handle log in in backend [1]
 - Authorisation [1]
- **Navigation bar [1]**
 - Not signed in: [2]
 - Home, Sign up/Log in, About,
 - (Might start with having everything visible when not logged in)
- **Home page [1]**
 - Title with under title
 - “PHOTO GRID – remember together” [1]
 - Small photo grid with happy images [2]
 - Sign up button [1]
- **Sign up/log in [1]**
 - Form [1]
 - Displays sign up as default
 - Name, email, password text inputs [1]
 - Submit button [1]
 - Handle sign up in backend [1]
 - Use Regex to display strength of typed in password? [3]
 - Already a member? Log in button
 - Email, password text inputs [1]
 - Log in button [1]
 - Handle log in in backend [1]
- **About [2]**
 - Text about the project and the concept of site [2]
 - Embedded video showing how the site works [3]
- **The process of making the site – page? [3]**
 - Link to GitHub repos? [3]
 - Display this documentation? [3]
 - Write down thoughts throughout the project [1] and display them? [3]
- **Footer [2]**
 - Created by and year [2]
 - Links to LinkedIn, GitHub and my portfolio [2]

Signed in:

•Header [1]

- Header title with under title [1]
 - "PHOTO GRID - remember together" [1]
- Make text a clickable <a> returning to start page [1]
- Sign out button that redirects to home page when clicked [1]

•Navigation bar [1]

- Signed in: [2]
 - Home, Sign out, About, My grids, Connected grids [2]
- (Might start with having everything visible when not logged in)

•My grids [1]

- Title -> "Hi X. Y. these are your grids"/"Hi X. Y. You haven't created any grids yet" [1]
- Create new grid-button [1]
 - Generate some sort of string (like an accesstoken/password?) that can be shared to other users to access the grid room. [1]
 - Not completely sure how to do this in the backend?
- List of all grids the user has created [1]
 - Delete button for each grid? [3]
 - A user can only delete a grid it has created!
 - Handle delete in backend [3]

•Connected grids

- Title -> "Hi X. Y. these are your connected grids"/"Hi X. Y. You haven't connected to any grids yet" [1]
- Add new grid (like an input field where you add the accesstoken? [1]
- Maybe a link that you can share with others? If you access the link, you are granted access to the grid?
- List of all grids user is connected to [1]
 - "Leave grid" button for each grid? [3]
 - Handle leave grid in backend [3]
- Link to each grid [1]

•Grid page [1]

- Title with the name of the grid [1]
- Images displayed in a grid as background of the page [1]
 - Some sort of fallback image and text if no images has been uploaded
 - Chat like section to write _or_ a comment section for each image?[3]
 - A variation could be to track how many times the "report" button has been pressed, then show how many times it has been reported in the frontend
- Upload image [1]
 - Drop zone? [2]
 - Upload several images at once? [3]

•Footer [2]

- Created by and year
- Links to LinkedIn, GitHub and portfolio

Models backend:

• User

- Name: String, req, minlength 1
- Email: String, req, minlength 3
- Password: String, req, minlength 6
- List of all grids created by user: List of objects(?) [{}]
- List of all grids user is connected to: List of objects(?) [{}]

• Grid

- Name of grid eg) Dwight Schrute's halloween costumes: String, req, minlength 5
- Password or String that gives access to grid room: String, req, minlength 10
- Name of user that created the grid, connected by: _id
- List of all images?

• Image

- img_counter for moderating: Number, default: 0
 - \$inc counter, when counter reaches 3 the images does not get displayed in the frontend [2]
- _id of the grid it's connected to: String
- List of all comments posted to that img: List of objects (?)
[{postedBy: user._id, comment: "Prison Mike looks really cool in this photo"}, {postedBy: otherUser._id, comment: "I think he looks dumb"}] [3]

• Comment (this should maybe be inside another model?)

- Name of user who commented
- Comment, String, req, minlength 5

TECH TO BE USED:

- **Backend:**

- Postman
- Mongoose
- MongoDB
- Heroku
- Node.js

- **Frontend:**

- React
- Redux
- Styled components (making it easier to further down the road make a react native application as another pet project)

- **Bothend:**

- Visual Studio Code
- Git
- The Hippo Team
- Technigos stack overflow team
- Code-a-longs on Mondays and Wednesdays
- Trello

TEAM MEMBERS:

Me, Christina P.

-_(`ツ)`_/-