

# Live Q&A App

## Doelgroep

- studenten die lessen volgen
- bezoekers die naar een presentatie komen kijken in de Creativity Gym (vd IMD cafe)
- bezoekers van een conferentie / workshop

## Doel

- het doelpubliek kan vragen stellen en op elkaars vragen antwoorden geven
- op die manier kan er per presentatie / les een document uitgebouwd worden waarbij vragen gesteld worden en antwoorden gegeven worden

## De gebruiker

- kan zich registreren/inloggen (mag via bijvoorbeeld passport.js)
- kan een nieuwe discussie starten en de link hiervan delen met anderen (deze persoon wordt moderator)
- hou de geolocatie van de discussie bij zodat anderen de discussie kunnen bekijken “in de buurt”
- kan vragen stellen in een discussie
- kan antwoorden geven op vragen
- links naar afbeeldingen worden automatisch getoond als afbeeldingen
- kan discussies “in de buurt” makkelijk terugvinden in je applicatie
- berichten moeten live bijgevoegd worden (via web sockets!), m.a.w. zodra iemand een reactie plaatst bij een vraag moet die reactie bij iedereen live op het scherm verschijnen

## De moderator

- Dit is de persoon die de discussie gestart heeft
- kan commentaren verwijderen
  - Indien een commentaar verwijderd wordt, moet die bij iedereen op het scherm *live* verdwijnen
- kan een discussie beëindigen of “op slot” zetten zodat er geen nieuwe commentaren of wijzigingen doorgevoerd kunnen worden

## De frontend / routes

- /
  - Dit is de homepage waarop je discussies *in de buurt* kan zien
  - Je kan doorklikken naar een discussie, inloggen of een discussie aanmaken
- /discussion/:id

- Detailpagina van één discussie
- Hierop zijn alle comments te zien en kan je comments plaatsen
- /discussion/create
  - Formulier om een nieuwe discussie te maken

### Voorbeeld van een discussie

A Web Page

# Les node.js


Waarvoor dient die expressjs

Volgens mij zit daar routing standaard in, anders moet je dat zelf schrijven

Klopt, thanks!

Wat is socket.io?

Da's een library waarmee je live updates kan sturen naar verschillende browsers



## Technische vereisten

- Werk vanaf dag 1 met GIT, de commit-history is bewijs van inzet
- Hou een google docs bij waarin staat wie wat gedaan heeft op welke datum
- Werkt via een API-based setup waarmee je communiceert via Javascript (kies zelf voor *vanilla* javascript, jQuery, angularjs, vuejs of een ander framework)
- Commentaren of nieuwe berichten in een discussie moeten *live* bijgevoegd worden bij alle lezers tegelijkertijd (zoals in Google docs)
- toon live welke en hoeveel gebruikers er in een discussie zijn bijgevoegd
  - Vb: door kleine profielfoto's te tonen
- gebruik web sockets (via socket.io bijvoorbeeld)
- gebruik SASS om uw CSS te schrijven
- discussies moeten achteraf in hun geheel opvraagbaar zijn, zorg dus voor een persistente data storage via mongodb
- zorg voor een aantrekkelijke UI
- gebruik gulp.js als build tool
  - CSS minification + concatenation
  - JS minification + concatenation
  - Image optimization
  - + kies nog één bijkomende gulp task
- werk met node.js + express + mongodb als basis

Je mag je applicatie komen verdedigen zodra ze klaar is, je hoeft dus niet te wachten tot de laatste les week. Klaar is klaar!

Werk in teams van twee studenten. Zorg dat elk teamlid **van in het begin** (geen excuses) met GIT werkt in een gedeelde GIT-repository. Elk teamlid moet evenveel node.js schrijven én kunnen verdedigen. Als een teamlid bijvoorbeeld enkel maar CSS of HTML schrijft zorgt dat automatisch voor een onvoldoende op de opdracht.

Vul de team-lijst hier aan:

<https://docs.google.com/spreadsheets/d/1NSiKdyg-jMTPnHqCTHDSmETGOB67rrrPWIPNpAJei04/edit?usp=sharing>

Veel succes!