

project-tech

Team

lab 6/8

today

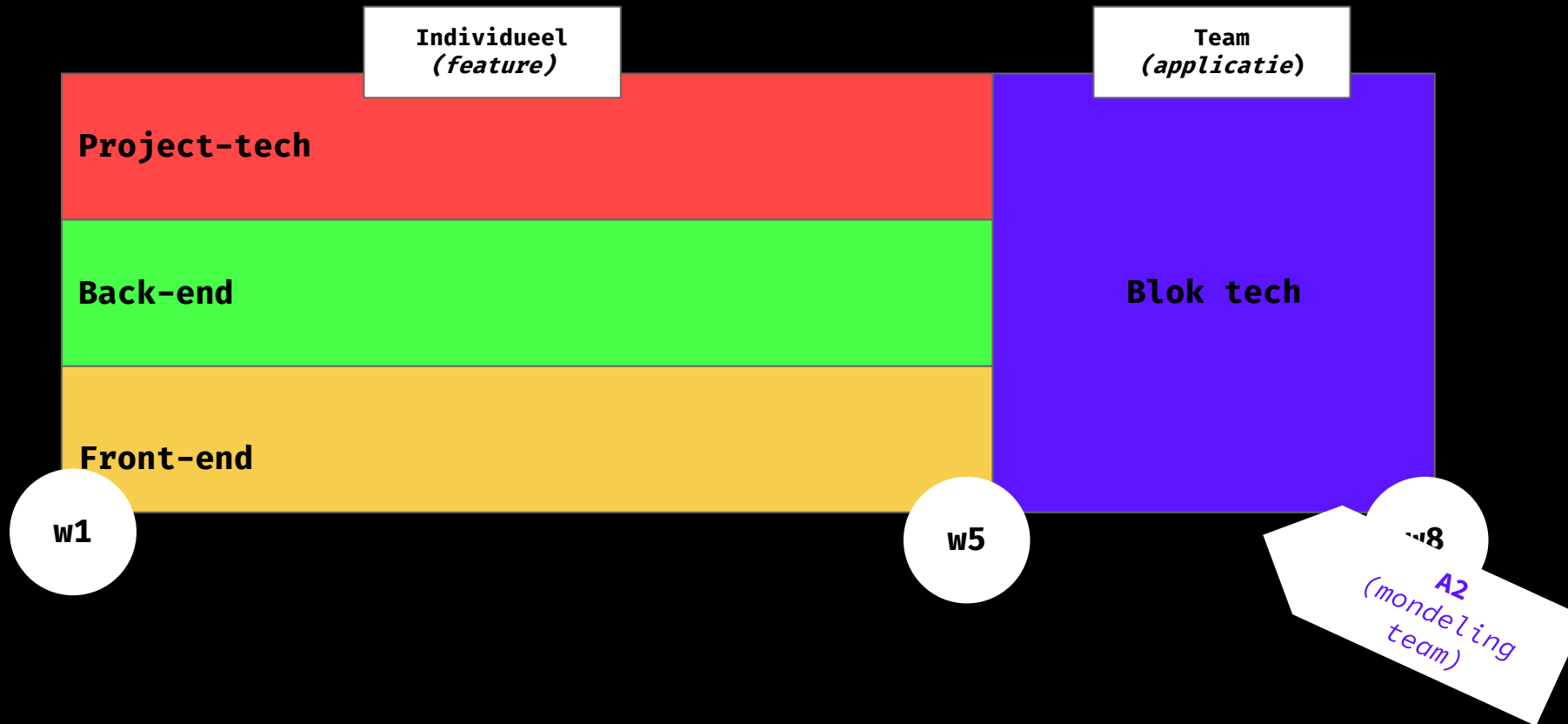
I.Recap

II.GitHub Flow

III.Team Stand-ups



Recap



Project-tech

- Samenwerking **op GitHub** (code, docs)
- **GitHub repo** zelf (configuratie, readme)
- **Coding standaarden**
- **Prototype** van de matching app

	1-2	3-4	5-6	7-8	9-10
Collaboration	The team doesn't use GitHub	The team uses version control but not all social features of GitHub	The team uses version control and the social features of GitHub to work on the project	The team worked with branches, pull requests and issues to collaborate easily	Each team member has equal contributions and tasks are delegated
Evaluation	Team members didn't review any code	Team members looked at some code but made no comments	Team members reviewed each other's code and processed issues and bugs other people assigned to them	Team members reviewed code and actively refactor code to make the application more coherent	Team members also have complex technical discussions and comments of
Application	The application doesn't work; there are errors and warnings	The application works but is incomplete	The team created a working dynamic prototype of a matching application	The application is advanced and is technically complex; it's well designed visually	The application is fantastic and the feature is complex. The team took special care of the interface and the user
Quality	The project is handed in on time, working without technical problems, and on GitHub	The code is readable, consistent and the code, project, and process are partially documented	Code adheres to standards; docs cover the process and what the project is and does; a live version is deployed	Code is good and docs are more useful and professional	Code is well structured logically

Samenwerken

Evalueren

Maken

Brightspace inleveren

Aan het eind dus inleveren;

- ❖ Link naar de **teamrepo (applicatie)**
- ❖ Link naar **documentatie (wiki)**
- ❖ Een video naar de **screencast presentatie**

Als vervanging voor
een 'presentatie' in
het klaslokaal

The image shows a development environment with three main components:

- Web Browser (Chrome):** Displays the registration page at `localhost:8000/registration`. The page has a form with the following sections:
 - What is your current education or work spot?** with a text input containing "Teacher of the year".
 - What are your movie preferences?** with a dropdown menu showing "#actionMovies" selected.
 - A red **Create** button.
 - A **Back to sign in** button.
- Code Editor:** Shows the `register.js` file with the following code:

```
currentUsers.js JS register.js x
routes > JS register.js > createAccount

35 next(err);
36 }
37 }
38
39 async function createAccount(req, res, next) {
40 // Page to fill in all user information to register
41 try {
42   const allUsers = await usersCollection.find().toArray();
43   let totalCount = allUsers.length + 1;
44   const password = req.body.password;
45   // hashes the password with salt
46   let hashPassword = bcrypt.hashSync(password, 10);
47   // body pulled from form
48   let firstName = req.body.firstName;
49   let lastName = req.body.lastName;
50   let email = req.body.email;
51   let gender = req.body.gender;
52   let age = req.body.age;
53   let work = req.body.work;
54   let movie1 = req.body.movie1;
55   let movie2 = req.body.movie2;
56
57   // makes age integer
58   let photo;
59   age = parseInt(age);
60   if (gender === 'man') {
61     photo = 'man.png';
62   } else {
63     photo = 'woman.png';
64   }
65
66   // data send to the DB
67   let data = {
68     'id': totalCount,
69     'firstName': firstName,
```
- Video Call:** A window showing three participants in a video conference.
- Terminal:** Shows the message: "Logged in as Gerard. Your session is destroyed. You can log in again to use the application."

GitHub Flow

🕒 5 minute read 📄 Download PDF version

The diagram illustrates a path through a network. A horizontal blue line at the top represents a timeline or sequence. Below it, a series of nodes (circles) are connected by edges (lines). The nodes are arranged in a sequence, with some nodes having additional connections. A path is highlighted in blue, starting from a purple node on the left, moving right, then down, then right again, and finally up to a purple node on the right. A dashed line indicates an alternative path, starting from a purple node on the left, moving right, then down, then right again, and finally up to a purple node on the right. The nodes are connected by a series of lines, with some nodes having additional connections. The path is highlighted in blue, and the dashed line indicates an alternative path.

Create a branch




Learning Lab

For Organizations

[Sign in](#)

Managing merge conflicts

 The GitHub Training Team

Merge conflicts are a normal part of working in Git. Learn why they happen and how to resolve them with ease.

[Start free course](#)

Join 12046 others!



Merge conflicts happen when two people make changes to the same file on GitHub—a common occurrence when you're working with other developers. While resolving differences might involve some discussion, merge conflicts don't have to be scary.

This course guides you through the steps to finding the best merge conflict solution, so your team can keep building.

What you'll learn

We'll answer common questions like:

- What is a merge conflict and what causes them?
- How do you resolve merge conflicts?
- How do you reduce merge conflicts?

Tags

[Git](#) [Merge conflicts](#) [GitHub](#)

Share *Managing merge conflicts*



Average time to complete

31 minutes

Free

All public courses on Learning Lab are *free*.

 [Get help](#)

[Post on the GitHub Community Forum](#)

Latest release

November 20, 2020



Search or jump to...



Pulls Issues Trending Explore



cmda-bt / pt-course-20-21 ✓

Watch ▾

Star

36

Fork

<> Code

Issues 33

Pull requests

Actions

Wiki

Settings



Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Default branch

The default branch is considered the "base" branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

main



Branch protection rules

Add rule

Define branch protection rules to disable force pushing, prevent branches from being deleted, and optionally require status checks before merging. New to branch protection rules? [Learn more.](#)

No branch protection rules defined yet

Filters ▾

Clear current search query, filters, and sorts

1 Open ✓ 16 Closed

Add CSS to authentication files
#38 by HappyPantss was closed on 14 Apr 2020 updated on 14 Apr 2020
code style

Only
#32 by enh was closed on 14 Apr 2020 updated on 14 Apr 2020
enhancement

Indentation of the Javascript
#26 by StanBankras was closed on 14 Apr 2020 updated on 14 Apr 2020
code style

Dashboard page
#30 by StanBankras was closed on 13 Apr 2020 updated on 13 Apr 2020
enhancement

Full profile preview page
#31 by StanBankras was closed on 13 Apr 2020 updated on 13 Apr 2020
enhancement

Connect filter feature with match slider users
#35 by StanBankras was closed on 13 Apr 2020 updated on 13 Apr 2020
enhancement

Voorbeeld van
samenwerken

Pull requests Issues Trending Explore

StanBankras / amatch

<> Code ⓘ Issues 1 ⚙ Pull requests 0 ⚙ Actions ⚙ Projects

Watch ⌵ ☆

Topic: Split route logic

Stan Bankras edited this page on Apr 7 · 6 revisions

Backend topic: Split route logic

This topic was worked on by Stan Bankras

A team project can become quite a mess if everyone works in the same server.js or index.js file: the file quickly adds a lot of lines and it takes long to read and understand what is happening. That's why I wanted to separate as much as possible

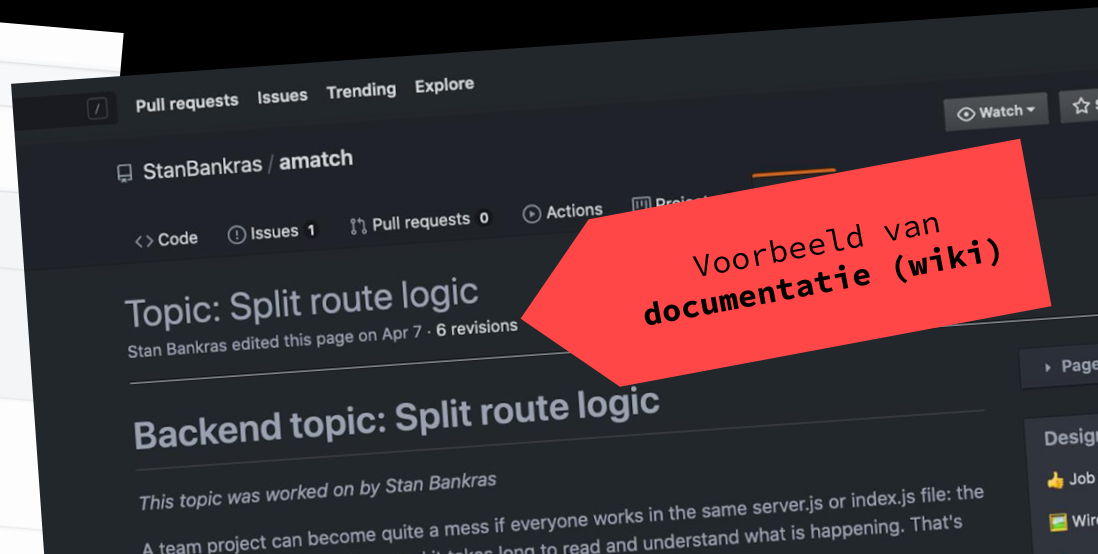
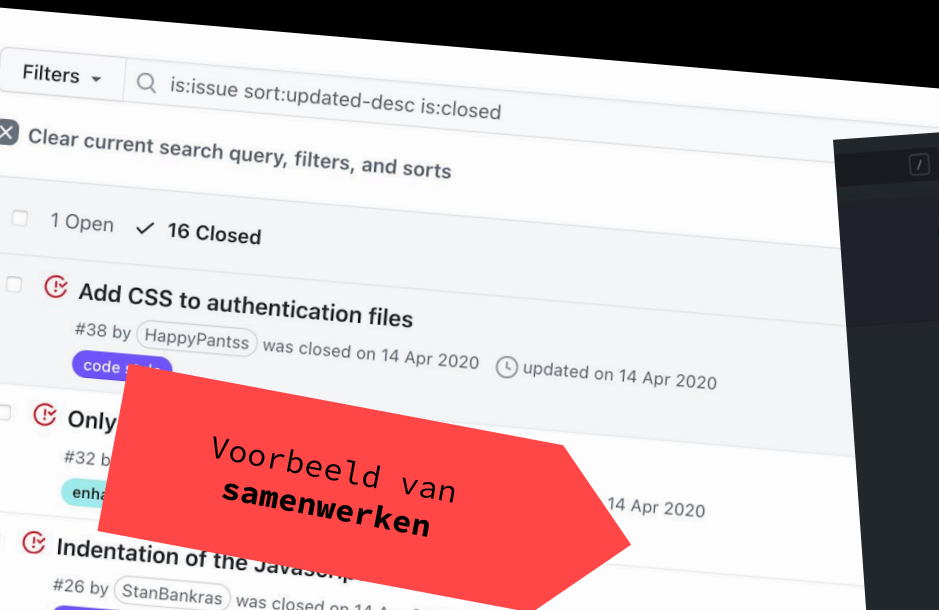
- Moving route logic to other files than the server
- Exporting the database connection from one central place

Moving route logic to other files than the server

In Javascript you can require other javascript files using require(). This can be used to import different files that host routes in the server.js to clear up some space:

```
const matchRouter = require('./routes/matches');
const chatRouter = require('./routes/chatting');
const authentication = require('./routes/authentication');
```

Voorbeeld van
documentatie (wiki)



Clearly document and show your contributions. Through wikis, commits, issues etc. You explain to us what you did.

Teams Standup

Team coaching



MVP



exit;

see you in lab-7!