COOKPIT PRESENTATION

TINF22B6 Semester 4

Karim Gharbi, Domenik Nedele,
Robin Dieser, Simon Hajek

TABLE OF CONTENTS

01

OUR FOCUS

Semester 3 and our start in semester 4

02

TECHNICAL IMPLEMENTATION

We want to give you a big overview about our technical aspects of the project

03

RUNNING DEMO

We want to share our progress and workflow with you

04

LESSONS LEARNED

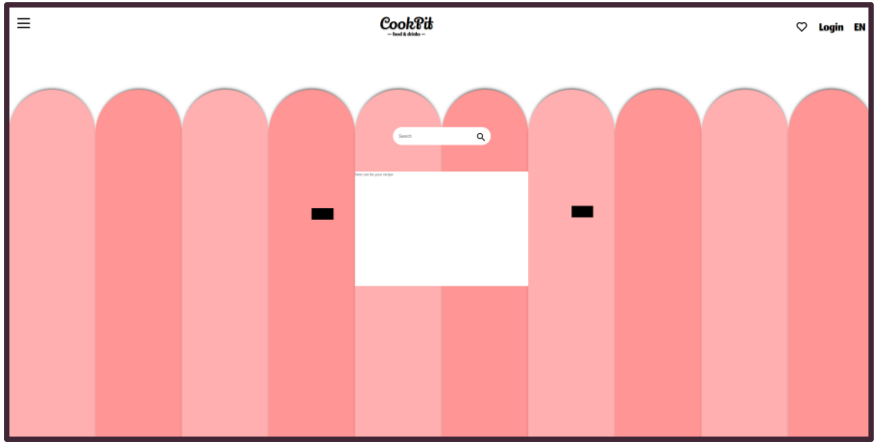
In this semester we got a lot of insights into software development and learnt many things

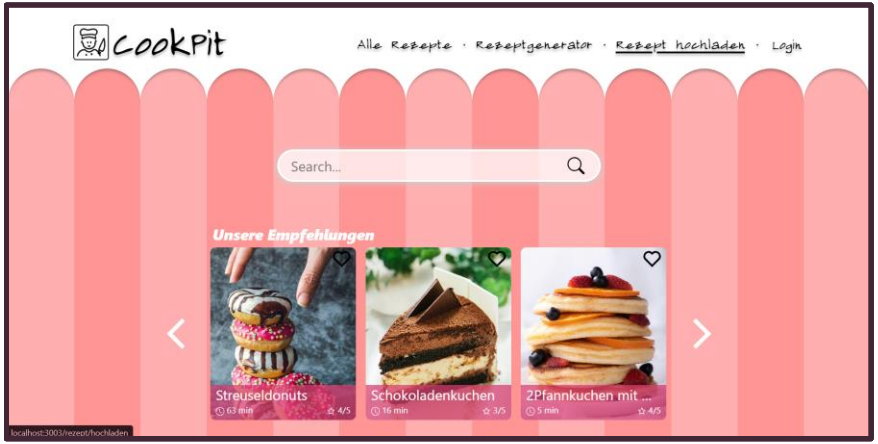
01

Recap

We have decided to keep CookPit as simple as possible:
➔ the reason for this: usability

We decided to not offer many special functions for the user, because we knew that only 10% of CookPit users actually use them. Therefore we made the decision to focus more on UX





OUR SELLING POINT

CookPit is a cooking platfrom where you can browse for recipes and upload your own

-> this topic were our main focus on

Userbility

full range of functions on all platforms

Efficiencies

fast and standardised communication between user and database

Simplicity

easy to find your way around and no hidden functions

02

TECHNICAL IMPLEMENTATION

Project setup

React as main framework

GitHub actions for automated tests

SQLite as Database

Express

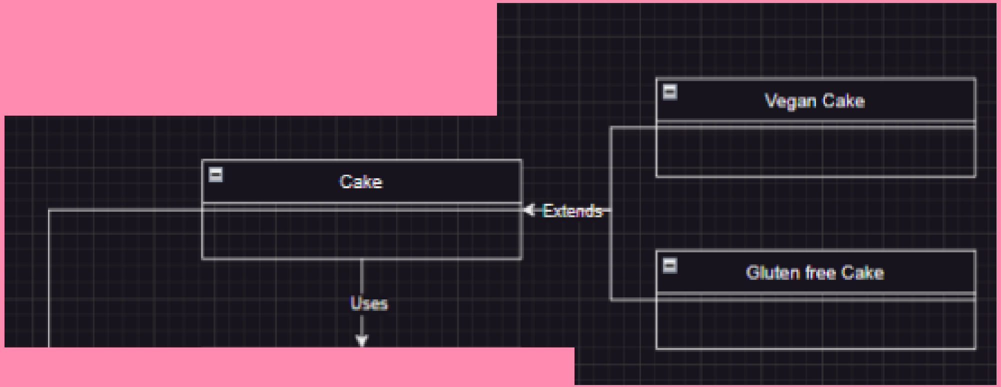
Architecture Design

Seperation between frontend, backend and database

Component-based architecture enables a modular and scalable solution, simplifying maintenance and future expansions. By clearly separating frontend, backend, and database, each component can be developed and optimized independently.

02

TECHNICAL IMPLEMENTATION



Design Patterns

Minimal use of the builder pattern
Problem:
-> CookPit is not object-orientated, but template for displaying a cake recipe is used for other categories

METRICS AND TESTS

Used Metrics:

- Web Application Performance Metrics (User Satisfaction / Apdex Scores)

  -> Apdex = (StatisfiedCount + (0.5 * ToleratingCount) +(0 * FrustratedCount))/TotalSamples

- Web Application Monitoring Tools

  - > Check user interactions

- Code Complexity Metric

  -> Callculate Code Complexity

Test Setup:

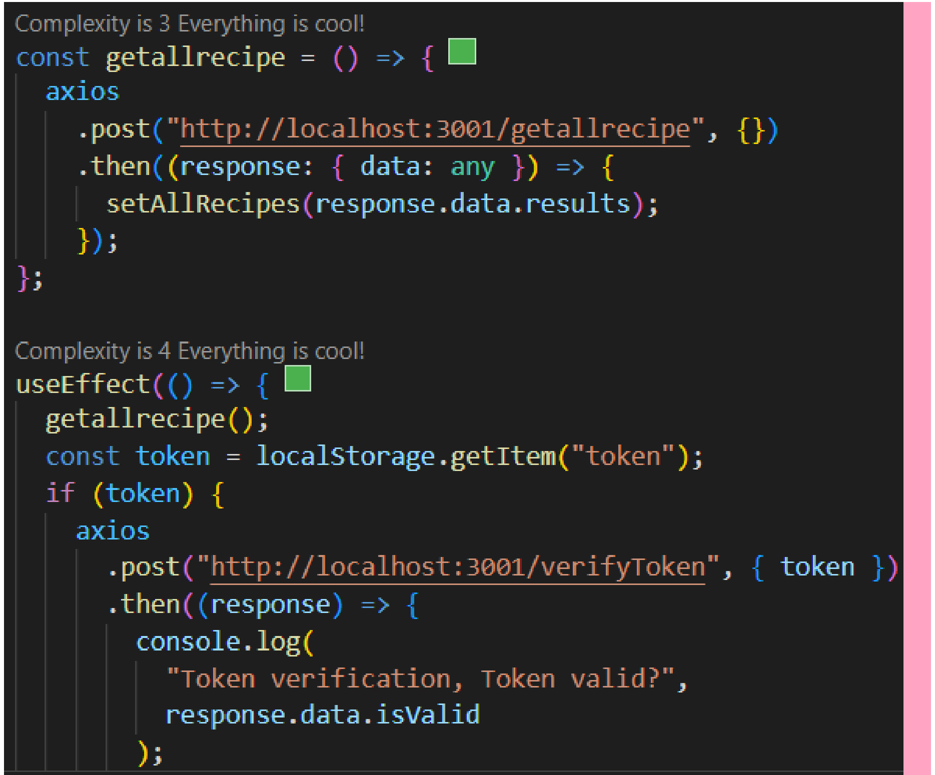
- Component Testing

  -> Key features like OpenAI, RecipeUpload, Communication between

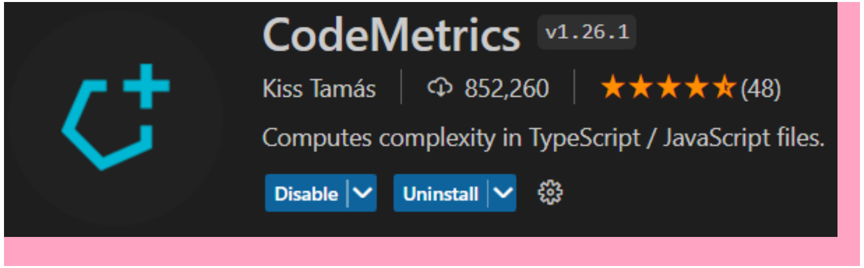
- Unit Testing:

  -> Test for Buttons, Inputfileds, Renderring

We have used a VSCode extention to callculate our Code complexity and establish Clean Code



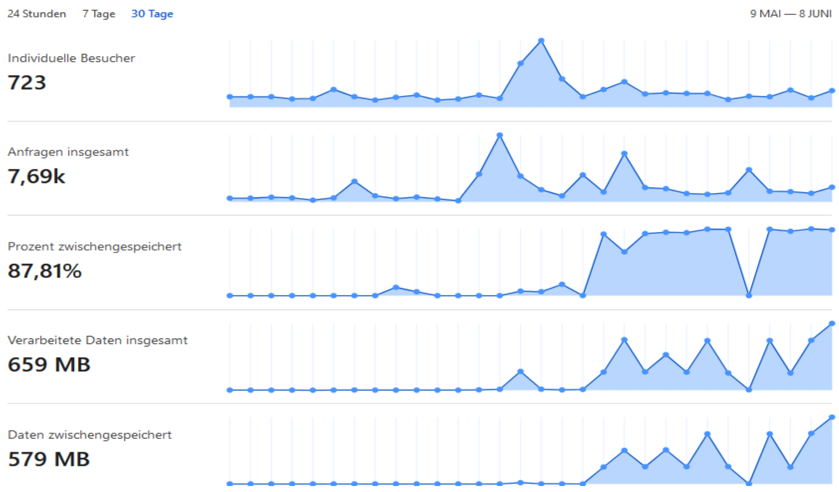
```
Complexity is 3 Everything is cool!
const getallrecipe = () => { 🟩
  axios
    .post("http://localhost:3001/getallrecipe", {})
    .then((response: { data: any }) => {
      setAllRecipes(response.data.results);
    });
};

Complexity is 4 Everything is cool!
useEffect(() => { 🟩
  getallrecipe();
  const token = localStorage.getItem("token");
  if (token) {
    axios
      .post("http://localhost:3001/verifyToken", { token })
      .then((response) => {
        console.log(
          "Token verification, Token valid?",
          response.data.isValid
        );
```



CodeMetrics v1.26.1

Kiss Tamás | 852,260 | ★★★★☆ (48)

Computes complexity in TypeScript / JavaScript files.



24 Stunden 7 Tage 30 Tage

9 MAI — 8 JUNI

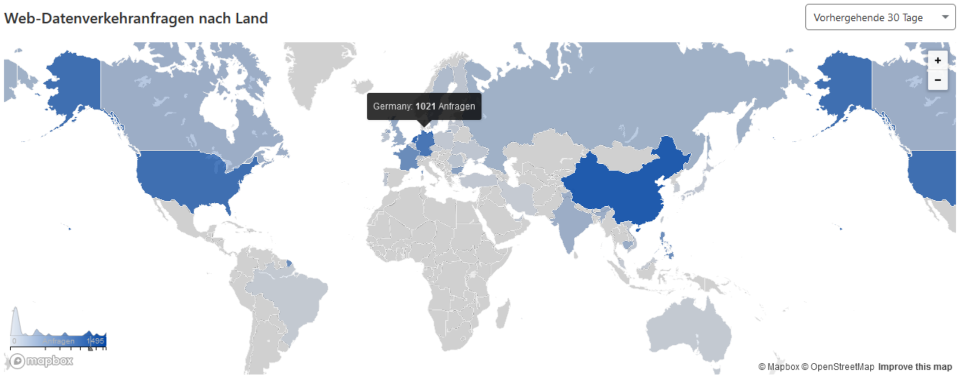
Individuelle Besucher
723

Anfragen insgesamt
7,69k

Prozent zwischengespeichert
87,81%

Verarbeitete Daten insgesamt
659 MB

Daten zwischengespeichert
579 MB



Web-Datenverkehranfragen nach Land

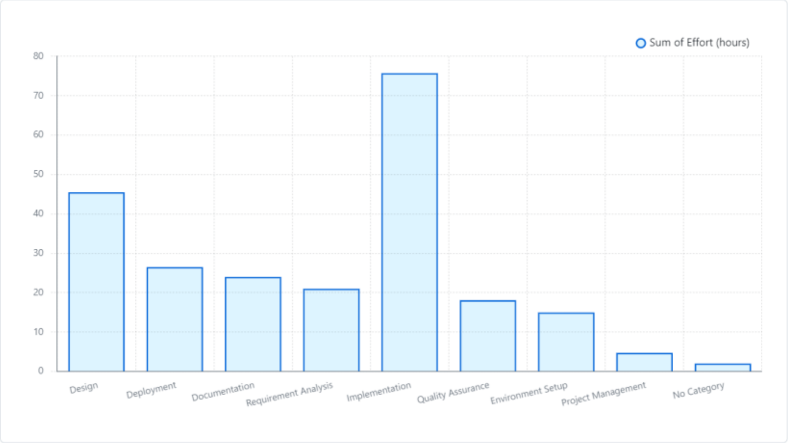
Vorhergehende 30 Tage

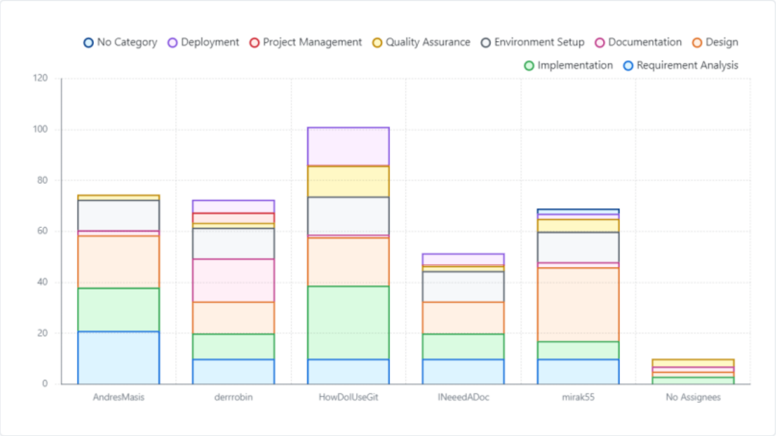
Germany: 1021 Anfragen

Web Application Monitoring Tools

We are using Strato fo hosting our web application and Mapbox to track the user interactions.

Last month we had in total 8000 users visting our website. We are looking forward to it





EFFORT

Chart one shows the planed time effort per category.

The second chart is displaying our planed time effort per person.

It can be said that the planning of the time was very difficult, that it was our first own project and we had not yet dealt with many technologies and therefore put a lot of time into the research

CI/CD Setup

Docker & NodeJS

Like other Teams we are using Github Actions for automated tests. For example NodeJS to ensure that CookPit runs on all possible NodeJS versions

Demo time

Follow the project updates on our weekly GitHub blog.

LESSONS LEARNED

04

WHAR WE HAVE LEARNED

Setting up a project

Planning

Framework and technologies

Teamwork

Data structure

Testing & Metrics