



TÍMOVÝ PROJEKT - ČO VARITĚ?

Brezina, Baláž, Onofrej, Beneda, Cibulka

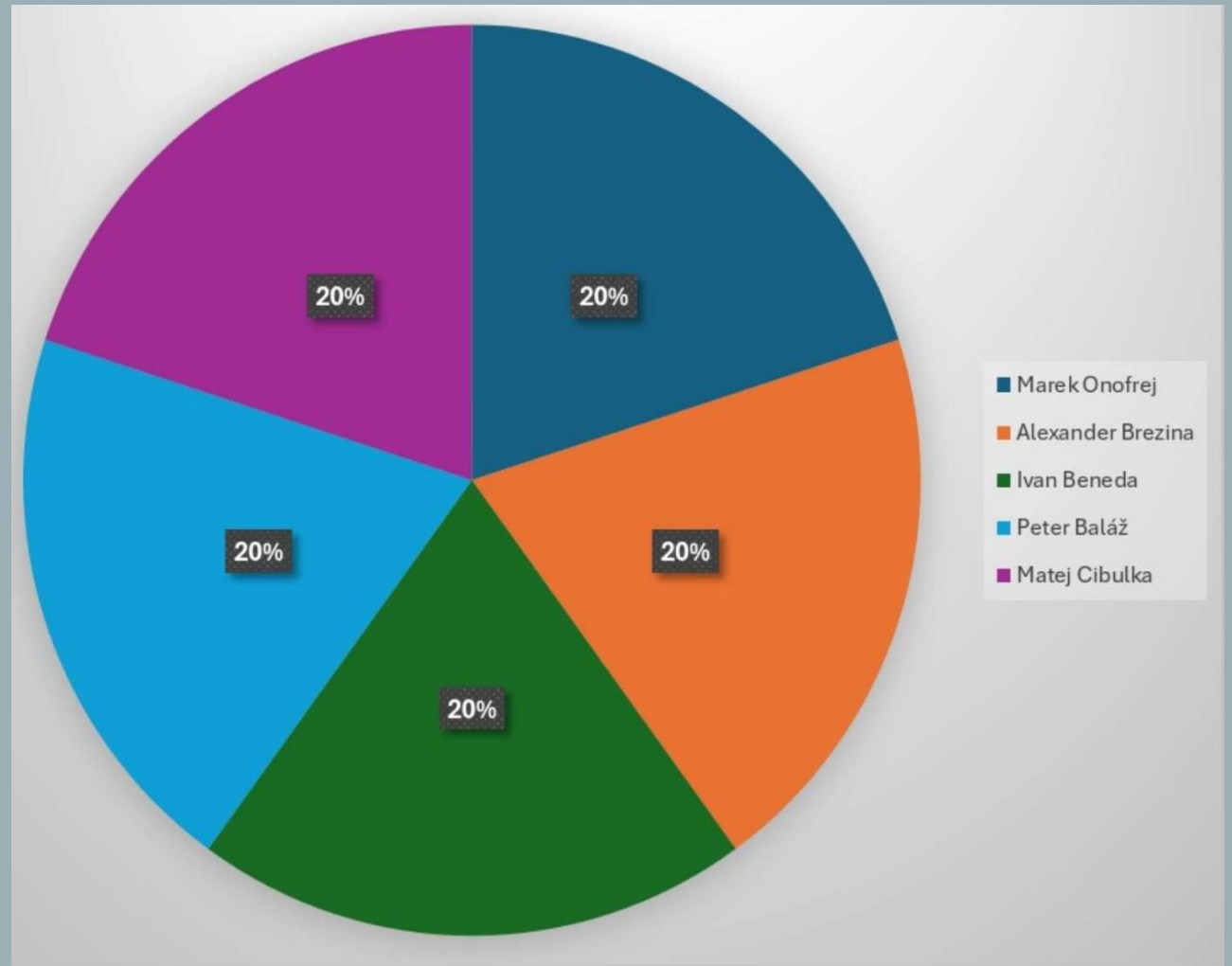
ČO SME ROBILI?

Webová aplikácia na recepty a ich
filtrovanie podľa surovín s možnosťou
náhodného výberu

AKTUÁLNY STAV PROJEKTU

● 90%~

PODIEL PRÁCE ČLENOV



BREZINA - VEDÚCI

Dohľad nad tímovou komunikáciou

Stanovovanie cieľov

Dohľad nad plnením úloh

Návrhy realizácie funkcií

Implementácia pridania receptu (FE)

 .vs Tvoja commit správa

 app Merge pull request #10 from timovyprojek

 lib POST,GET,DELETE

 public Tvoja commit správa

 .gitignore init

 README.md init

 eslint.config.mjs init

 jsconfig.json init

 next.config.mjs init

 package-lock.json Tvoja commit správa

 package.json fixes

 postcss.config.js fixes

 postcss.config.mjs fixes

 tailwind.config.js Add files via upload

BENEDA – MAIN FRONTEND

Front end layout

Komunikácia s backend (DB)

Podstránky na prepojenie

Vytváranie nových funkcionalít

Covarit

Zoznam

+ Pridať recept

Recepty

Náhodný recept

Hľadať podľa názvu alebo ingrediencie...



Párance s ricottou

Klikni na detail pre zobrazenie ingrediencií a postupu.

Detail

Zmazať

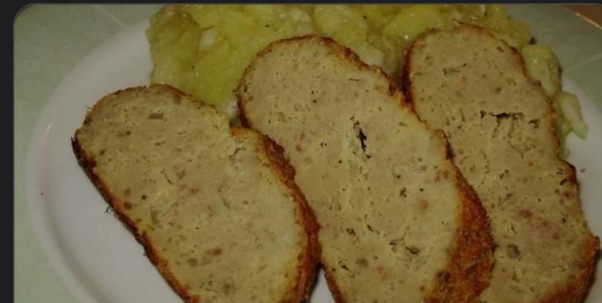


Slaný cukinový koláč

Klikni na detail pre zobrazenie ingrediencií a postupu.

Detail

Zmazať



Sekaná z bôčika

Klikni na detail pre zobrazenie ingrediencií a postupu.



Francúzske zemiaky s údeným tofu (pri diéte s obmedzením tuku)

CIBULKA – SUPPORT FRONTEND

Hlavný dizajn

Tailwind

Bugfixes/opravy

Wireframes



Kuracia pečeň na hubách a smotane

Ingrediencie:

- 800 g kuracia pečeň
- 1 väčšia konzerva (šampiňóny) huby
- 3 ks cibuľa
- 250 ml smotana na varenie
- 250 ml mlieko
- olej olivový
- soľ
- paprika červená mletá
- korenie
- na zahustenie múka
- majorán

Postup:

- Vo väčšom hrnci speniť na olivovom oleji nadrobno pokrájanú cibuľu s hubami.
- Pridať pokrájanú pečeň a dusiť v šťave čo pustí cca 15 min.
- Poprášiť mletou červenou paprikou a korením. Nechať ešte pár minút variť.
- Pridať smotanu rozmiešanú s mliekom.
- Dovaříť na miernom ohni nejakých 10 min. Ak by bola omáčka stále riedka zahustiť múkou, ale aby sa nezrazila. (Ja si odoberem trochu omáčky do taniera, pomiešam s múkou a cez jemné sitko nalejem späť do hrnca.) Nakoniec dochutiť majoránkou a môže sa podávať. Mne najlepšie chutí s knedlou, ale môže byť aj ryža, alebo zemiaky.

← Naspäť

🗑 Zmazať

ONOFREJ – SUPPORT BACKEND

Napojenie aplikácie na databázu

MongoDB

Implementácia základných CRUD

operácií (POST, DELETE)

Overovanie správnosti uloženia a

mazania dát

```
import { MongoClient } from "mongodb";

const uri = process.env.MONGODB_URI;
if (!uri) {
  throw new Error("Please define the MONGODB_URI environment variable inside .env.local");
}

const dbName = process.env.MONGODB_DB || "stranka";

let cached = global._mongo || (global._mongo = { conn: null, promise: null });

export async function connect() {
  if (cached.conn) return cached.conn;
  if (!cached.promise) {
    const client = new MongoClient(uri);
    cached.promise = client.connect().then((client) => {
      return { client, db: client.db(dbName) };
    });
  }
  cached.conn = await cached.promise;
  return cached.conn;
}

export async function getDb() {
  const { db } = await connect();
  return db;
}

export async function DELETE(request) {
  try {
    const url = new URL(request.url);
    const id = url.searchParams.get("id");
    if (!id) return NextResponse.json({ error: "Missing id" }, { status: 400 });

    const { db } = await connect();
    const result = await db.collection("recepty").deleteOne({ _id: new ObjectId(id) });
  } catch (err) {
    return NextResponse.json({ error: String(err) }, { status: 500 });
  }
}
```

BALÁŽ – MAIN BACKEND

Implementácia API

Implementácia vyhľadávania a
filtrovaní

Implementácia náhodného receptu

Opravovanie nefunkčných vecí
ostatných členov tímu

```
export async function GET(request) {
  try {
    const { db } = await connect();
    const collection = db.collection("recepty");

    const url = new URL(request.url);
    const page = parseInt(url.searchParams.get("page") || "1", 10);
    const perPage = parseInt(url.searchParams.get("perPage") || "10", 10);
    const search = url.searchParams.get("search") || "";

    let query = {};
    if (search) {
      const terms = search
        .split(",")
        .map((s) => s.trim())
        .filter((s) => s.length > 0);

      if (terms.length === 1) {
        const t = terms[0];
        query = {
          $or: [
            { title: { $regex: t, $options: "i" } },
            { ingredients: { $regex: t, $options: "i" } },
          ],
        };
      } else if (terms.length > 1) {
        query = {
          $and: terms.map((t) => ({
            $or: [
              { title: { $regex: t, $options: "i" } },
              { ingredients: { $regex: t, $options: "i" } },
            ],
          })),
        };
      }
    }

    let docs, total;
    if (!search) {
      [docs, total] = await Promise.all([
        collection.aggregate([
          { $sample: { size: perPage } }
        ]).toArray(),
        collection.countDocuments({}),
      ]);
    } else {
      [docs, total] = await Promise.all([
        collection
          .find(query)
          .sort({ createdAt: -1 })
          .skip((page - 1) * perPage)
          .limit(perPage)
          .toArray(),
        collection.countDocuments(query),
      ]);
    }

    const recipes = docs.map((d) => ({ ...d, _id: d._id.toString() }));
    return NextResponse.json({ recipes, total });
  } catch (err) {
    return NextResponse.json({ error: String(err) }, { status: 500 });
  }
}
```

ĎAKUJEME ZA POZORNOST!