FİNAL PROJE RAPORU

Dönem: Güz 2026

Ders: Web Tasarımı ve Programlama

Proje Başlığı: The Sweet Lab

Ad: Şevval

Soyad: ARSLAN

Numarası: 22040301030

Bölümü: Yazılım Mühendisliği

E-posta: sevvalarslan@stu.topkapi.edu.tr

Web Sitesi Linki:

https://thesweetlab-sevval.onrender.com/

(Lütfen site açılırken birkaç saniye bekleyiniz.)

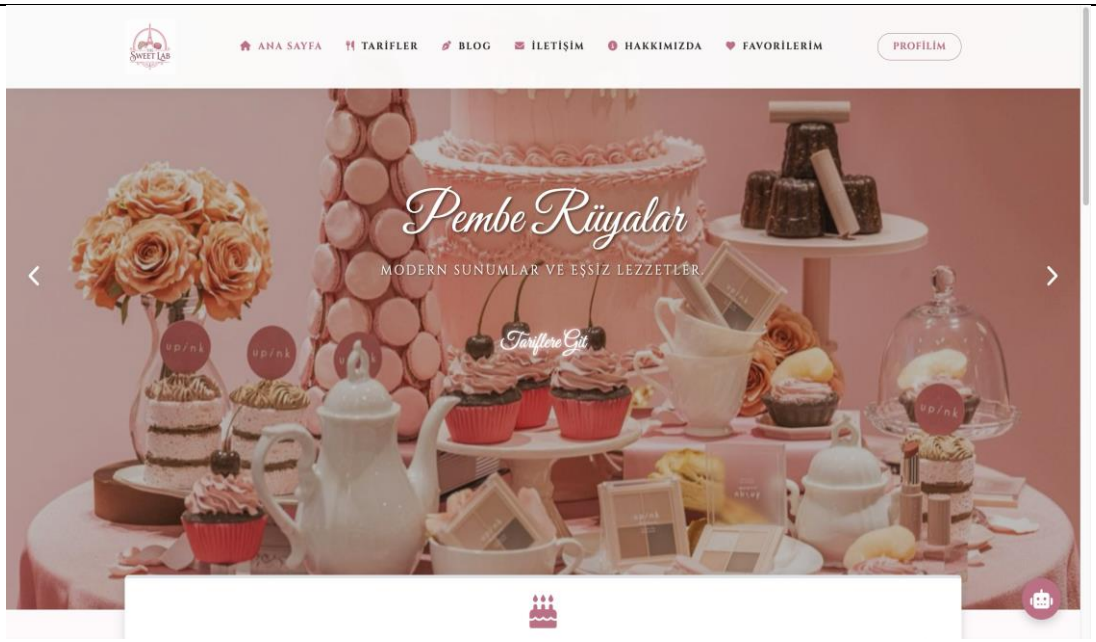
GitHub Repo Bağlantısı:
https://github.com/iamsevval/TheSweetLab

A) GİRİŞ

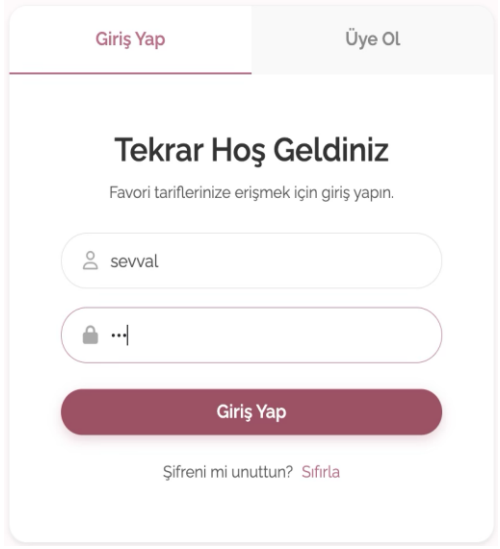
The Sweet Lab, kullanıcıların tatlı tariflerini paylaşıp etkileşime girebildiği, **Yapay Zekâ (AI) destekli** dinamik bir web platformudur. Proje, sunucu tarafında **Node.js** ve **Express.js** mimarisi üzerine inşa edilmiş olup, veri kalıcılığı için **SQLite** veritabanı kullanılmıştır.

Kullanıcı arayüzünde (Frontend) **HTML5, CSS3 ve JavaScript** ile birlikte **Bootstrap** kütüphanesi kullanılarak responsive (mobil uyumlu) bir deneyim sağlanmıştır. Proje kapsamında temel **CRUD** (Create, Read, Update, Delete) prensiplerinin yanı sıra; **Google Gemini AI** entegrasyonu ile tarif analizi, **Session** tabanlı güvenli kimlik doğrulama ve dinamik içerik yönetimi gibi ileri seviye web teknikleri uygulanmıştır. Kod incelemesi bölümünde, projenin tüm kaynak kodları yerine, sistemin ana fonksiyonlarını temsil eden önemli parçalar derlenerek rapora eklenmiştir.

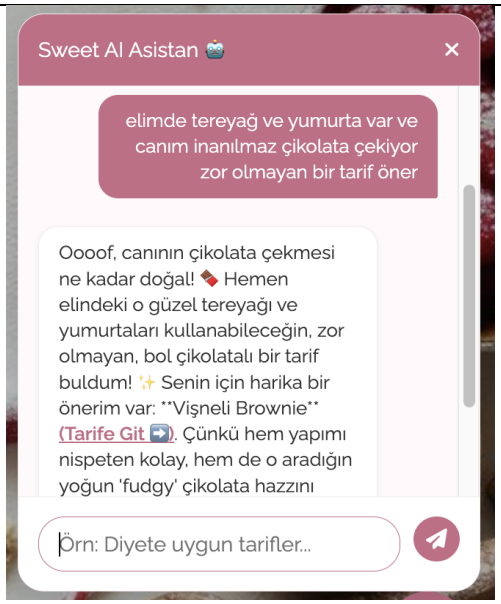
B) SAYFA GÖRSELLERİ VE İLGİLİ KODLAR



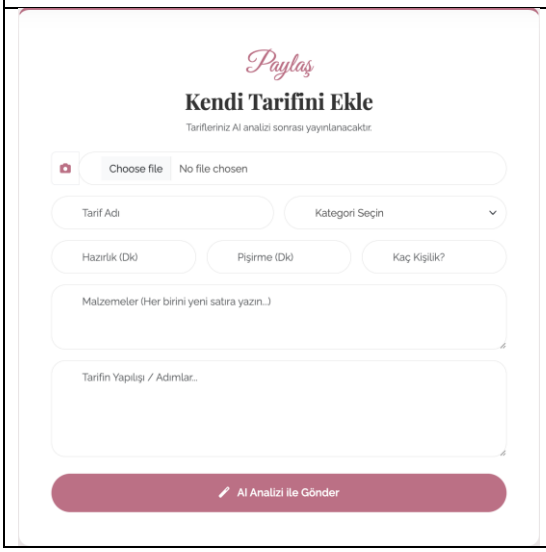

```
<section class="hero-slider">
    <div class="slide active">
        <img src="../assets/images/kapak2.jpg" class="slider-bg">
        <div class="overlay"></div>
        <div class="slide-content">
            <h1>Pembe Rüyalar</h1>
            <p>Modern sunumlar ve eşsiz lezzetler.</p>
            <a href="recipes.html" class="btn btn-magic mt-4">Tariflere Git</a>
        </div>
    </div>
    <div class="slide">
        <img src="../assets/images/anaekran3.jpg" class="slider-bg">
        <div class="overlay"></div>
        <div class="slide-content">
            <h1>Sanatın En Tatlı Hali</h1>
            <p>The Sweet Lab mutfağında Fransız esintileri.</p>
            <a href="blog.html" class="btn btn-magic mt-4">Keşfetmeye Başla</a>
        </div>
    </div>
    <div class="slide">
        <img src="../assets/images/kapak3.jpg" class="slider-bg">
        <div class="overlay"></div>
        <div class="slide-content">
            <h1>Yapay Zeka Destekli Tatlar</h1>
            <p>Damak tadınıza uygun akıllı öneriler.</p>
        </div> </div>
```




```
/* --- Giriş Kontrolü (routes/authRoutes.js) --- */
router.post('/login', (req, res) => { const { username, password
} = req.body;

// 1. Kullanıcıyı Veritabanında Ara
const sql = "SELECT * FROM users WHERE username = ?";


db.get(sql, [username], (err, user) => {
    if (err || !user) {
        return res.send('<script>alert("Kullanıcı
bulunamadı!");</script>');
    }


    // 2. Şifreyi Karşılaştır (Bcrypt ile Güvenli Kontrol)
    const isMatch = bcrypt.compareSync(password, user.password);


    if (isMatch) {
        // 3. Giriş Başarılı: Oturum Bilgilerini Kaydet
        req.session.userId = user.id;
        req.session.username = user.username;...
        res.redirect('/profile.html');
    } else {
        // 4. Hatalı Şifre
        res.send('<script>alert("Hatalı şifre!");</script>');
    }
  });
});
```




```
const { GoogleGenerativeAI } = require("@google/generative-ai");
const genAI = new GoogleGenerativeAI(process.env.API_KEY);


router.post('/api/chat', async (req, res) => {


const userMessage = req.body.message;


// 1. Adım: Veritabanındaki tarifleri çek (RAG Mimarisi)
const sql = "SELECT id, title, ingredients, category FROM
recipes";
db.all(sql, [], async (err, recipes) => {...
    // 2. Adım: AI Modelini Başlat
    const model = genAI.getGenerativeModel({ model: "gemini-
flash-latest" });
    // 3. Adım: Sisteme Rol ve Veri Verme (Prompt)
    const prompt = `
    Sen "Sweet AI" adında tatlı dilli bir pastacı asistanısın.
    ELİNDEKİ TARİFLER: ${JSON.stringify(recipes)}
    KULLANICI MESAJI: "${userMessage}"
    GÖREV: Kullanıcının malzemelerine uygun tarifi bul ve öner.
    `;
    // 4. Adım: Cevabı Üret ve Gönder
    const result = await model.generateContent(prompt);
    res.json({ reply: result.response.text() });...
   });

});
```




```
<form id="recipeForm">
    <input type="file" id="r-image" class="form-
control" required>
    <input type="text" id="r-title" class="form-
control" placeholder="Tarif Adı">

    <select id="r-category" class="form-select">
        <option value="diyet">Diyet & Hafif
Tatlılar</option>
        <option value="cikolatali">Çikolatalı</option>
        </select>...

    <textarea id="r-ingredients" rows="3"
placeholder="Malzemeler..."></textarea>

    <textarea id="r-desc" rows="4"
placeholder="Adımlar..."></textarea>

    <button type="submit" class="btn btn-auth w-100">
        <i class="fas fa-magic"></i> AI Analizi ile
Gönder
    </button>
</form>
```

Yulaflı ve Orman Meyveli Sağlıklı Kurabiye

25 Dk | 6 Kişilik

Eline Sağlık 3 | Ben de Yaptım 4

Tarifi İncele

DIYET

```
recipes.forEach(r =>{

container.innerHTML += `

       <div class="card-body text-center p-4">
            <h3 class="recipe-title h5 fw-bold">${r.title}</h3>

            <div class="d-flex justify-content-center gap-2 mb-3">
                <button class="btn btn-interaction"
onclick="sendInteraction(${r.id}, 'like', this)">
                    <i class="far fa-thumbs-up"></i>
                    <span>Eline Sağlık</span> <b>${r.like_count ||
0}</b>
                </button>
                <button class="btn btn-interaction"
onclick="sendInteraction(${r.id}, 'cooked', this)">
                    <i class="fas fa-hands-clapping"></i>
                    <span>Ben de Yaptım</span> <b>${r.cooked_count
|| 0}</b>
                </button>
            </div>
            <button class="btn btn-outline-pink w-100 rounded-pill"
onclick="openRecipePreview(${r.id})">
                Tarifi İncele
            </button>
        </div>
    </div>
</div>`;
});
```

Fransız Usulü Kruvasan

145 DK | SWEET LAB ŞEFİ

Malzemeler 5

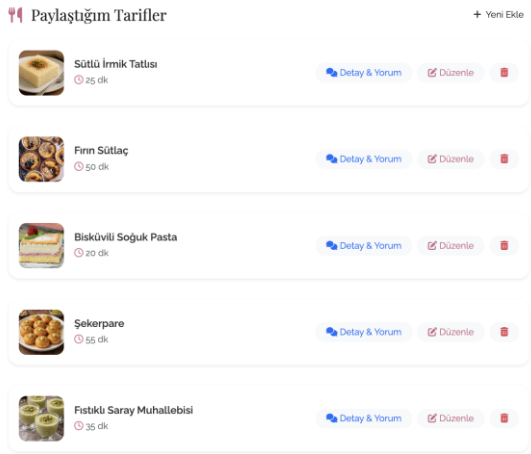
- 500 gram un (tercihen yüksek proteinli ekmek unu)
- 60 gram toz şeker
- 10 gram tuz
- 10 gram kuru maya
- 150 ml soğuk süt

Hazırlanışı

Ön Hamur: Un, şeker, tuz, maya, süt, su ve 50g tereyağını yoğurun. Pürüzsüz bir hamur elde edince streçleyip buzdolabında en az 2 saat (ideali bir gece) dinlendirin.

Tereyağ Bloğu: 250g soğuk tereyağını iki yağlı kağıt arasına koyun ve merdaneyle vurarak yaklaşık 15x15 cm boyutlarında bir kare haline getirin. Tekrar dolaba koyun.

```
window.updateServings = (change) => { const servingsElement =
document.getElementById('current-servings'); if (!servingsElement) return;


// Mevcut porsiyonu al ve güncelle
let current = parseInt(servingsElement.innerText);
let newPortion = current + change;


if (newPortion < 1) return;
servingsElement.innerText = newPortion;


// Listedeki tüm malzemeleri yeni porsiyona göre güncelle
document.querySelectorAll('.ingredient-amount').forEach(span => {
    const baseAmount = parseFloat(span.dataset.baseAmount);
    const originalServings =
parseInt(span.dataset.originalServings);


    // Formül: (Miktar / Eski Porsiyon) * Yeni Porsiyon
    const calculated = (baseAmount / originalServings) *
newPortion;


    // Sonucu ekrana yaz
    span.innerText = Number.isInteger(calculated) ? calculated :
calculated.toFixed(1);
  });
};
```




```
recipes.forEach(r => { list.innerHTML += `
    <div class="card shadow-sm border-0 mb-3">
        <div class="card-body d-flex justify-content-between align-
items-center p-3">
            <div class="d-flex align-items-center">
                <img src="${r.image_url}" ...>
                <div class="ms-3">
                    <h6 class="fw-bold">${r.title}</h6>
                    ...
                </div>
            </div>
            <div class="d-flex gap-2">
                <a href="..." class="btn ...">Detay</a>

                <button onclick="openEditModal(${r.id})" class="btn
...">
                    <i class="fas fa-edit"></i> Düzenle
                </button>
                <button onclick="confirmDelete(${r.id})" class="btn
...">
                    <i class="fas fa-trash-alt"></i>
                </button>
            </div>
        </div>
    </div>`;  });
```

```
// routes/recipeRoutes.js

// TARİF EKLE (CREATE)
router.post('/add-recipe',
upload.single('image'), (req, res) => {
    // Kullanıcı giriş yapmış mı kontrolü
    if (!req.session.userId) {
        return res.status(401).json({ error:
"Önce giriş yapmalısın!" });
    }

    const { title, description, ingredients,
category, prep_time, cook_time, servings } =
req.body;
    // Resim varsa yolunu al, yoksa varsayılanı
kullan
    const imageUrl = req.file ?
`/uploads/${req.file.filename}` :
'/assets/images/default-cake.jpg';
    const userId = req.session.userId;

    const sql = `INSERT INTO recipes (title,
description, ingredients, category, prep_time,
cook_time, servings, image_url, user_id)
                 VALUES (?, ?, ?, ?, ?, ?, ?, ?,
?)`;

    db.run(sql, [title, description,
ingredients, category, prep_time, cook_time,
servings, imageUrl, userId], function(err) {
        if (err) return res.status(500).json({
error: "Veritabanı hatası!" });

        // Eklenen son ID'yi (this.lastID)
döndürerek başarılı yanıt veriyoruz
        res.json({ success: true, message:
"Tarif başarıyla eklendi!", new_recipe_id:
this.lastID });
    });
});
```

```
// routes/recipeRoutes.js

// TÜM TARİFLERİ GETİR (READ)
router.get('/recipes', (req, res) => {
    // Tarifleri ID sırasına göre (en yeni
en üstte) çekiyoruz
    const sql = "SELECT id, title,
description, ingredients, category,
prep_time, cook_time, servings, image_url,
user_id, like_count, cooked_count FROM
recipes ORDER BY id DESC";

    db.all(sql, [], (err, rows) => {
        if (err) return
res.status(500).json({ error: err.message
});
        // Tüm satırları JSON formatında
frontend'e gönderiyoruz
        res.json(rows);
    });
});
```

```
// TARİF GÜNCELLEME (UPDATE)
app.post('/api/update-recipe/:id',
uploadConfig.single('image'), (req, res) => {
    const recipeId = req.params.id;
    const { title, category, prep_time, cook_time,
servings, ingredients, description } = req.body;
    const userId = req.session.userId;

    // Güvenlik: Sadece giriş yapmış kullanıcı işlem
yapabilir
    if (!userId) return res.status(401).json({ success:
false, error: "Giriş yapmalısın." });

    let sql = `UPDATE recipes SET title=?, category=?,
prep_time=?, cook_time=?, servings=?, ingredients=?,
description=?`;
    let params = [title, category, prep_time,
cook_time, servings, ingredients, description];

    // Eğer kullanıcı yeni bir resim yüklediyse,
sorguya image_url alanını da ekle.
    if (req.file) {
        const imageUrl =
`/uploads/${req.file.filename}`;
        sql += `, image_url=?`;
        params.push(imageUrl);
    }
    // WHERE koşulu ile sadece ilgili tarif ve sahibi
güncellenir
    sql += ` WHERE id=? AND user_id=?`;
    params.push(recipeId, userId);

    db.run(sql, params, function(err) {
        if (err) return res.status(500).json({ success:
false, error: "Veritabanı hatası" });
        res.json({ success: true, message: "Tarif
güncellendi!" }); });});
```

```
// routes/recipeRoutes.js

// TARİF SİL (DELETE)
router.delete('/delete-recipe/:id', (req,
res) => {
    // Yetki kontrolü
    if (!req.session.userId) return
res.status(401).json({ error: "Yetkisiz
işlem!" });

    // Sadece ID yetmez, silen kişi o
tarifin sahibi mi (user_id) kontrolü de
yapıyoruz
    const sql = "DELETE FROM recipes WHERE
id = ? AND user_id = ?";

    db.run(sql, [req.params.id,
req.session.userId], function(err) {
        if (err) return
res.status(500).json({ error: err.message
});
        res.json({ success: true });
    });
});
```

`` const sqlite3 = require('sqlite3').verbose(); const db = new sqlite3.Database('./sweetlab.db', (err) => { if (err) console.error('Veritabanı hatası:', err.message); else console.log('SQlite veritabanına bağlanıldı.'); }); // ... (Tablo oluşturma kodları devam ediyor) module.exports = db; ```	**Teknik Not:** Veritabanı bağlantısı için SQLite3 sürücüsü kullanılarak asenkron bir bağlantı yapısı kurulmuştur. Kod bloğunda görüldüğü üzere, bağlantı hataları callback fonksiyonları ile yakalanarak (Error Handling) konsola raporlanmaktadır.
`` // OTURUM (SESSION) AYARI app.use(session({ secret: 'sweetlab_gizli_anahtar', resave: false, saveUninitialized: false, cookie: { secure: false, // Localhost için false maxAge: 1000 * 60 * 60 * 24 // 24 Saatlik oturum } })); ```	**Teknik Not:** Kullanıcı deneyiminin sürekliliği için sunucu tarafında oturum yönetimi yapılandırılmıştır. “cookie” ayarları üzerinden güvenli (secure) ve süreli (maxAge) oturum anahtarları oluşturularak, kullanıcının sayfalar arası geçişte tekrar giriş yapması engellenir.
`` // --- KAYIT OL (REGISTER) --- router.post('/register', (req, res) => { const { username, email, password } = req.body; // Şifreyi güvenlik için Hash'liyoruz const hashedPassword = bcrypt.hashSync(password, 10); const sql = "INSERT INTO users (username, email, password) VALUES (?, ?, ?)"; db.run(sql, [username, email, hashedPassword], function (err) { if (err) { // Hata yönetimi... return res.send('<script>alert("Bu kullanıcı zaten kayıtlı!"); window.location.href="/profile.html";</script>'); } res.send('<script>alert("Kayıt Başarılı! Lütfen Giriş Yapın."); window.location.href="/profile.html";</script>'); }); }); ```	**Teknik Not:** Veri güvenliği standartlarına (Cybersecurity Best Practices) uygun olarak kayıt sistemi geliştirilmiştir. bcrypt.hashSync(password, 10) fonksiyonu ile parola, 'salt' değeri eklenerek karmaşıklaştırılır. Bu işlem, veritabanı ele geçirilse dahi kullanıcı şifrelerinin çalınmasını önler.