FATİH SULTAN MEHMET VAKIF UNIVERSITY

WEB DESIGN AND PROGRAMMING PROJECT REPORT

FATIH SULTAN MEHMET VAKIF UNIVERSITY

https://foodtrendguide.tech

Students Name	: Rumeysa Sare Bayram
Student ID	: 2121251032
Project Topic	: A Website Project
Course Instructor	: Assoc. Prof. Dr. Samet Kaya

CONTENT

1) INTRODUCTION

FoodTrendGuide is a robust, full-stack web application designed to revolutionize how users discover and interact with dining venues. The platform serves as a centralized hub for food enthusiasts to explore trending restaurants, read community reviews, and curate personalized bookmark collections. By integrating local static data with dynamic external APIs, the project provides a comprehensive and interactive map-based experience for urban exploration.

2) SYSTEM ARCHITECTURE

The application follows a **Decoupled Client-Server Architecture**, ensuring scalability and a clear separation of concerns.

2.1 Backend Architecture (The Engine)

The backend is built using **Spring Boot 3.x**, following the "Controller-Service-Repository" pattern:

- **Controller Layer:** Defines RESTful endpoints for authentication, venue management, and user interactions.
- **Service Layer:** Contains the business logic, including complex integrations with **Google Places API** and **OpenStreetMap (OSM)** for real-time location data.
- **Data Access Layer (JPA):** Manages persistence logic through Spring Data JPA, mapping Java objects to the **MySQL** relational database.
- **Security:** Implements **Spring Security** to handle JWT-based authentication and Google OAuth2 integration, ensuring secure user sessions.

2.2 Frontend Architecture (The Interface)

The frontend is a modern **Single Page Application (SPA)** built with **React** and **TypeScript**:

- **State Management:** Utilizes React hooks for managing local state and component lifecycles.
- **Styling:** Leverages **Tailwind CSS** for a responsive, utility-first design approach.
- **Routing:** Managed by React Router to provide seamless navigation between pages like ExplorePage, VenueDetailPage, and ProfilePage.

3) TECHNOLOGY STACK

The **FoodTrendGuide** platform is engineered using a sophisticated and modern technology stack designed for high performance, security, and scalability. The core of the application is powered by **Java 21** and **Spring Boot 3.5.7**, providing a robust backend environment that leverages the latest features of the Java ecosystem. For the frontend, the project utilizes **React 18** paired with **TypeScript**, ensuring a type-safe and highly interactive user interface. The visual presentation is managed through **Tailwind CSS**, which allows for a responsive and modern utility-first design. Data persistence is handled by a **MySQL 8.0** relational database, with **Spring Data JPA** and **Hibernate** facilitating efficient object-relational mapping.

Security is a primary pillar of the architecture, implemented through **Spring Security** with support for **JWT-based authentication** and **Google OAuth2** for seamless social logins. The development lifecycle is streamlined using **Maven** for backend dependency management and **Vite** as a next-generation frontend build tool. Finally, the entire system is deployed on an **AWS EC2** instance running **Ubuntu**, where **Nginx** serves as a reverse proxy to manage HTTPS traffic and **Systemd** ensures the continuous availability of the backend service.

4) DETAİLED DİRECTORY STRUCTURE ANALYSİS

The project is organized into two primary modules:

4.1 Backend Module (/backend)

- **config/**: Contains security filters (SecurityConfig), CORS settings, and data loaders.
- **controller/**: Houses 11 specialized controllers, ranging from AuthController for user management to AdminController for system-level operations.
- **model/ & entity/**: Defines the data schema, including User, Venue, Review, and BlogPost entities.
- **service/**: Critical services like GooglePlacesService handle the external data ingestion.

4.2 Frontend Module (/frontend)

- **api/**: Contains Axios configurations (authApi.ts, venueApi.ts) for backend communication.
- **components/**: Reusable UI elements such as Navbar, SearchBar, and VenueCard.
- **pages/**: 13 distinct views catering to different user journeys, including a dedicated AdminSyncPage for data synchronization.

5) KEY FEATURES & IMPLEMENTATION

5.1 Dynamic Data Synchronization

The application features an administrative sync tool. The `AdminSyncPage` interacts with the `GooglePlacesService` on the backend to fetch the latest venue information, ratings, and photos directly into the local database.

5.2 Responsive Discovery

The `ExplorePage` and `FilterBar` allow users to sort venues by category and location, utilizing `turkiye-data.ts` for localized filtering.

5.3 Secure User Lifecycle

From registration to profile management, user data is protected. Recent updates involved migrating API calls from absolute paths (HTTP) to relative paths (`/api`) to comply with **SSL/TLS (HTTPS)** requirements, preventing Mixed Content vulnerabilities.

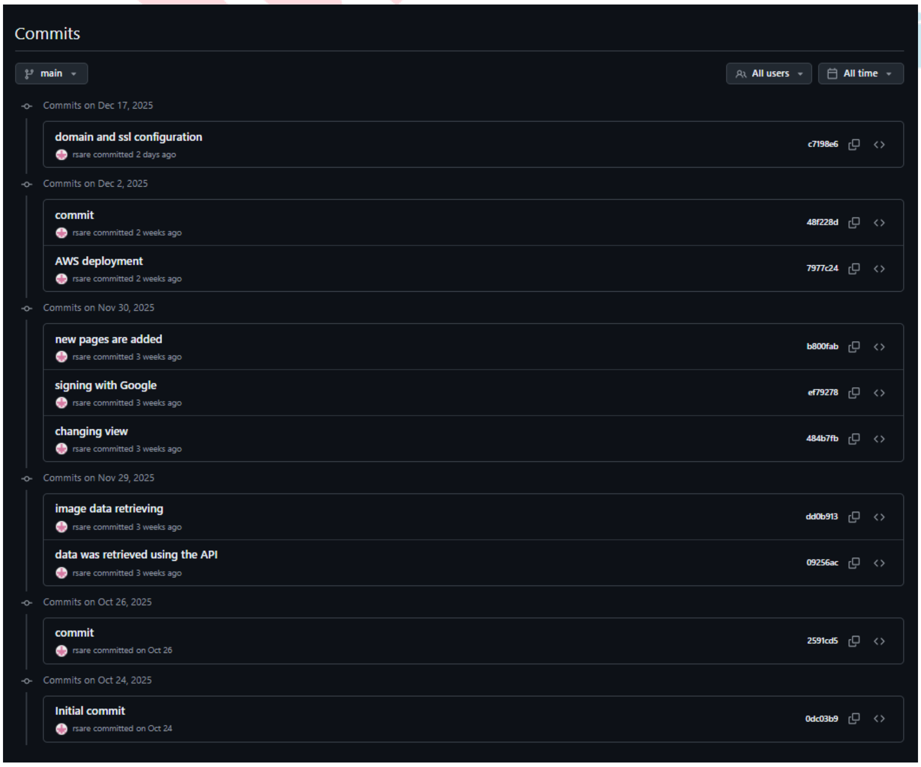
6) DEPLOYMENT & INFRASTRUCTURE

The application is deployed on an **AWS EC2** instance running Ubuntu.

- **Production Build:** The React application is built into static assets and hosted within the Spring Boot static/ directory, allowing the entire platform to run as a single executable .jar file.
- **Process Management:** The backend is managed as a **Systemd service** (foodtrend.service), ensuring high availability and automatic restarts.
- **Network Security: Nginx** acts as a reverse proxy, handling SSL termination and forwarding requests to the internal port 8080.

7) CONCLUSION

FoodTrendGuide demonstrates a successful integration of modern web technologies to solve real-world data discovery challenges. By combining the stability of Spring Boot with the agility of React, the platform provides a high-performance environment for users. The current infrastructure is ready for future enhancements, such as AI-driven restaurant recommendations and real-time social feed integrations.



Commits

Commits on Dec 17, 2025

domain and ssl configuration
rsare committed 2 days ago — c7198e6

Commits on Dec 2, 2025

commit
rsare committed 2 weeks ago — 48f228d

AWS deployment
rsare committed 2 weeks ago — 7977c24

Commits on Nov 30, 2025

new pages are added
rsare committed 3 weeks ago — b800fab

signing with Google
rsare committed 3 weeks ago — ef79278

changing view
rsare committed 3 weeks ago — 484b7fb

Commits on Nov 29, 2025

image data retrieving
rsare committed 3 weeks ago — dd0b913

data was retrieved using the API
rsare committed 3 weeks ago — 09256ac

Commits on Oct 26, 2025

commit
rsare committed on Oct 26 — 2591cd5

Commits on Oct 24, 2025

Initial commit
rsare committed on Oct 24 — 0dc03b9