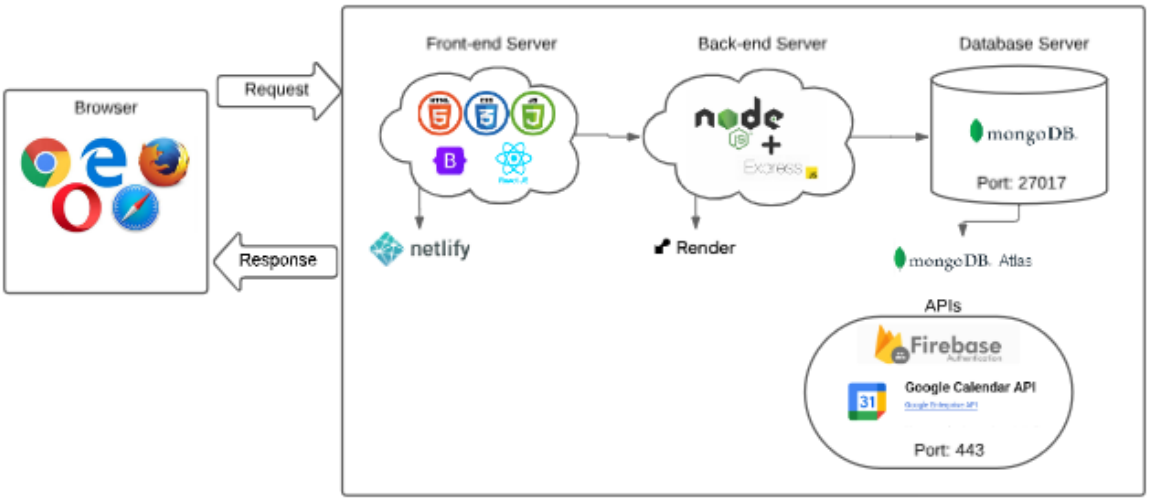
Web Application Architecture Design



Server and Port Configuration of the Web System

The system will be implemented using a **cloud-based distributed architecture**, composed of three main layers: **Front-end**, **Back-end**, and **Database**, each deployed on a separate server to ensure independence, scalability, and high availability.

1. Front-end (Presentation Layer)

- **Technology:** React.js for interface construction and Bootstrap only for styles management..
- **Server:** Netlify
- **Port:** No explicit port is used, as Netlify automatically assigns an HTTPS domain **(e.g., https://nombreproyecto.netlify.app)**.
- **Description:**

  This layer contains the graphical user interface developed with HTML, CSS, JavaScript, and React.js, enabling the creation of dynamic, reusable components. Bootstrap is used solely for visual design and adaptation to different devices.

  The front-end sends HTTP/HTTPS requests to the Back-end using libraries such as **axios** or **fetch**, ensuring efficient and secure communication with the business logic layer.

2. Back-end (Business Logic Layer)

- **Technology:** Node.js with the Express.js framework
- **Server:** Render
- **Port:** Automatically assigned by the Render server
- **Description:**

The Back-end handles business logic, processes requests from the Front-end, and communicates with the database via HTTPS. It is deployed on Render, a cloud service that allows continuous execution of Node.js applications with load balancing and automatic HTTPS support.

3. Database

- **Technology:** MongoDB managed with Mongoose
- **Server:** MongoDB Atlas
- **Port:** Internally uses 27017 (not publicly accessible; connection is made through the MongoDB URI)
- **Description:** MongoDB Atlas provides automatic replication, backup, and scalability, ensuring the availability and protection of data. The Back-end connects to the database using a secure connection string **(mongodb+srv://username:password@cluster.mongodb.net)**.

4. APIs

- **Google Calendar API**

Technology: Google Calendar API v3
Base URL: https://www.googleapis.com/calendar/v3
Port: 443 (HTTPS)
Purpose: Calendar integration for managing events and appointments
Description:
The Google Calendar API allows the system to **create, read, update, and delete events** in users' calendars. This integration facilitates scheduling, reminders, and synchronization with personal calendars.
The Back-end acts as an intermediary between the Front-end and the Google API, handling the **OAuth 2.0 credentials** required for authorization.

- **Firebase Authentication API**

Technology: Firebase Authentication (Google Identity Platform)
Base URL: https://identitytoolkit.googleapis.com/v1
Port: 443 (HTTPS)
Purpose: User authentication and access control
Description:
The Firebase Authentication API provides **secure authentication services** based on the Google Identity Platform infrastructure. This API uses **OAuth 2.0** and **OpenID Connect (OIDC)** standards for identity validation and generates **JWT tokens** to maintain active sessions.
The Back-end can act as an intermediary between the Front-end and Firebase, managing authentication tokens and validating access to the system's protected resources.