Week 1 Exploration (Mentor - Abinaya)

- Zainab Nusaiba

Objective:

1. What is a server? Why do we need servers?
2. What is backend work and frontend work?
3. What is Authentication (Google Authentication

1.1 What is a Server

A computer or a system is generally used to act as a server. It communicates with other servers known as clients over the network. Servers provide resources, programs, and various services to their clients.

Servers can also be considered large data storage and processing devices. There exist various types of servers, some of the most used ones include web servers, mail servers, virtual servers, file servers, print servers, database servers, DNS servers, game servers, and application servers.

It simply listens to the requests and it is present in the backend to provide the data that the client has requested.

1.2 Why do we need servers

Most of the basic and important tasks are done with the help of servers. Computers might store the data we need to access but servers store all the data available on the website and the data needed for its functioning.

This allows the users to avoid downloading the data to their local devices. Servers simply store the information and allow its access over a network such as the internet. This then makes the website accessible in various ways such as mobile devices, and desktops.

Some reasons why we need servers are
- They are more powerful than normal workstations.
- Enabling stronger security for the data.
- Enabling easy recovery and backup.
- Can be specially designed and built with more processing speed and storage capacity.

- It is more stable.
- Software provided for the protection of data for the server also protects the devices that are connected to the server over that network. Hence security against viruses and malfunctions is provided.

Some other reasons include its use as client-server model/architecture, data storage, virtual/cloud servers, in the gaming field, and more.

2.1 Frontend Work

The visible parts of the webpage that the client side can see are referred to as Frontend work. That is everything that the user can see such as a webpage, website, or app. That includes the data contents present in the webpage integrated with the way it is designed.

The programming languages used to control what the user will be interacting with are:

- HTML - provides structure and content
- CSS - provides style, basic interactivity, and animations
- JS - provides functioning for the website by adding interactivity and complex animations

2.2 Backend Work

It deals with those parts of a webpage, website, or app that the user can't see. It is complex functioning, integrations with different languages, content management, database storage, and more.

It is the development of server-side logic that boosts the website from behind the scenes

Small web apps can be created just with the help of javascript but for complex problems and huge applications, it will face performance issues. In such a case, backend languages are used. Such as:

- Ruby
- Python
- Java
- PHP

Backend work is ensuring that the various aspects of the backend are running smoothly and hand-in-hand. Those aspects are:

- Databases
- Application Programming Interfaces (APIs)
- Business logic

- Servers
- Integration
- Architectures

1.2 Google Authentication

Google Authentication refers to the two-factor application (2FA) process of verification. It is used for security and as a safety mechanism from intruder attacks. It is the verification for the account identification that the user persists in being the owner of. Its working mechanism is:

- User types their Username and password and the number for TOTP
- It provides a 6-digit TOTP (Time based One Time Password) that is sent to the user
- Passwords are changed every 30-60 seconds for further security

I have implemented a react application that is integrated with google Auth as a demo for its functioning.

1. User presses "Sign in with Google" and is redirected to Google log in screen

2. Google redirects user back to callback route, passing in an access code

3. Client sends the ID token to our API server

4. Our API server verifies and decodes the ID token, then searches for a matching user in our database or makes a new one.

The found user is added to the request session.

5. Server responds with success.

Future requests are authenticated with session cookie.

Our React client

Google's oAuth service

Our express REST API service