

ECE-2

Assignment - I
Web-deginingAdmno - 202SB0311018
Name - Ishakumari

3

Web

→ HTTP
and s

→ FTP:

4 Frame

→ Front
→ Back

5 Other

→ We
→ API
GptQ-2 Diff
devSolⁿ:
(i) Wor
sid
(ii) De
us(iii) HT
Rec(iv) Cr
be
(v) M
ne

- Q-1 Define Web Technologies. Explain different types of Web technologies with examples

Web technologies are the tools, language and techniques used for designing, developing, and running website or web applications.

Types:

1. Front-End Technologies (User Interface)
 → HTML (Hyper Text Markup Language): Define structure of web page's pages. Examples: - Creating headings, paragraphs, tables.
- CSS (Cascading Style Sheets): Add styling and layout. Examples: - Changing colours, fonts and page design.
- JavaScript: adds interactivity.
 Example: Form validation, dynamic menus

2. Back-End Technologies (server-side)

- PHP, Python (Django/Flask), Node.js, Java (Spring): used for server logic. Examples: Handling log in system, database queries.
- Database: MySQL, MongoDB, Oracle, Example: Storing user data in an e-commerce site.

3 Web protocols

- HTTP / HTTPS : For transferring data b/w Client and server.
- FTP : For file transfer

4 Frameworks & Libraries

- Frontend : React, Angular, Bootstrap.
- Backend : Express, Laravel

5 Other Technologies

- Web Servers : Apache, Nginx
- APIs (Rest, GraphQL) : For communication b/w apps.

Q-2 Difference b/w Front End developer and Back End developer.

Solⁿ:

Front-End

- (i) Work on the client-side (what user sees)
- (ii) Designs and develops user interface (UI) using HTML, CSS, Javascript, React, Angular.
- (iii) Creates responsive and attractive web pages.
- (iv) Making buttons, forms, navigation menu.

Back End

- (i) Works on servers side (what user doesn't see)
- (ii) Manages database servers and application logic
- (iii) PHP, Python, Java, Node.js, SQL, MongoDB.

- (iv) Ensures data usage, storage, security, and functionality
- (v) Processing login, storing user data, payment system

a-3 Explain Client Server architecture with diagram.

Client - Server Architecture is a model where multiple client (user) request services and a central server provides those services.

(i) Client :- → The user's device (computer, mobile, browser)
→ Sends request to the server (e.g. Open a webpage, login)

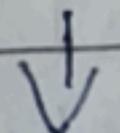
(ii) Server : → Powerful computer that stores data, applications and resources

(iii) Communication : → Take place through protocols like HTTP/HTTPS

Diagram →

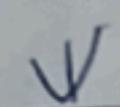
User Laptop / Mobile

↑ Request



↓
[Server]

↑ Query / Data



↓
[Database]