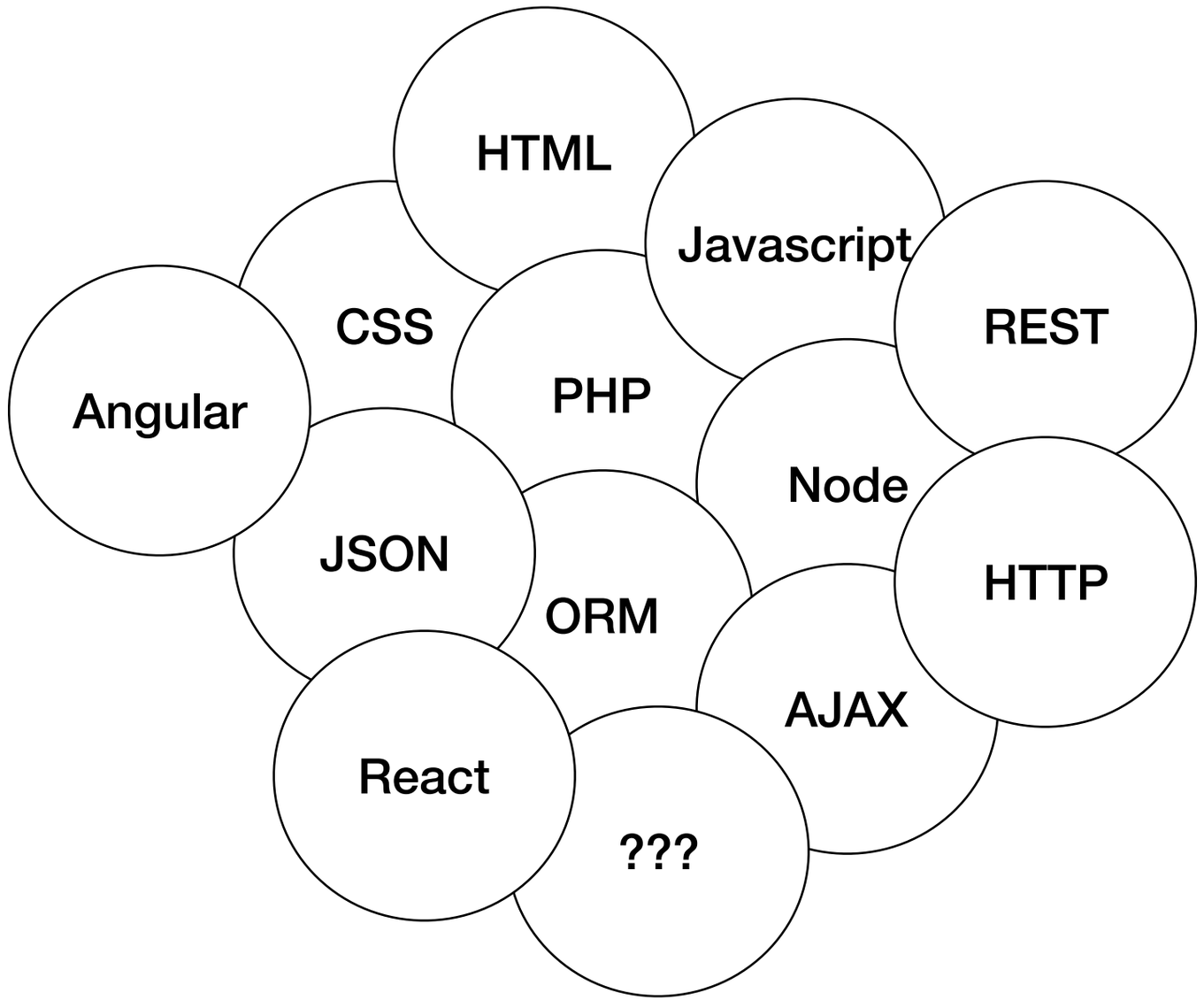
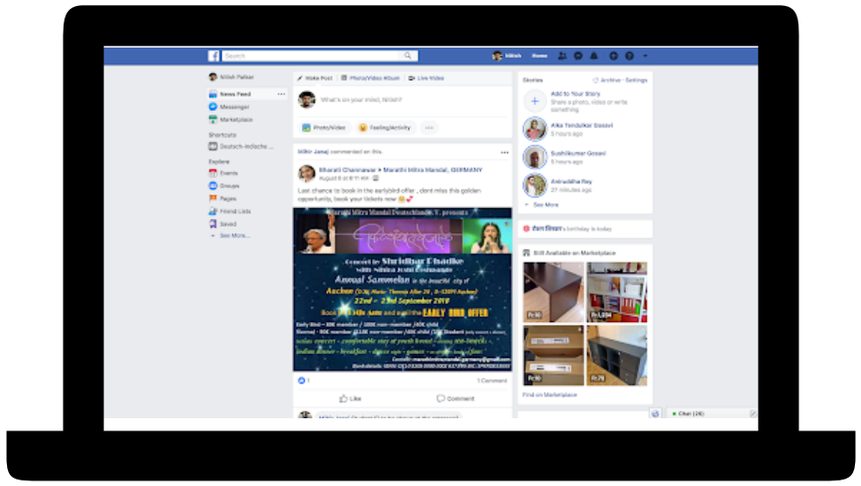
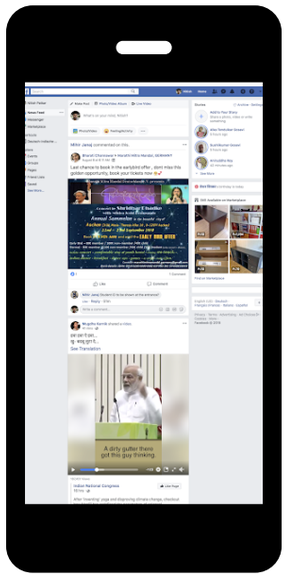
A brief introduction to web development

ESE 2019 Exercise getting started

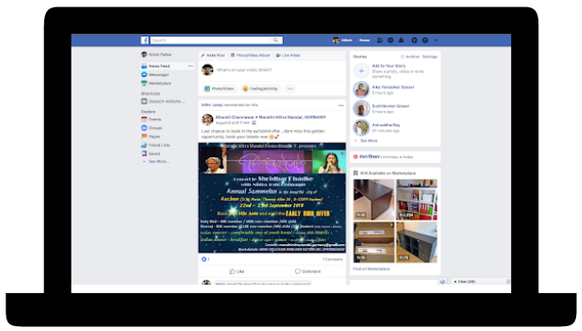

HTML
Javascript
REST
CSS
PHP
Angular
Node
JSON
ORM
HTTP
AJAX
React
???

Simplified view on today’s web development





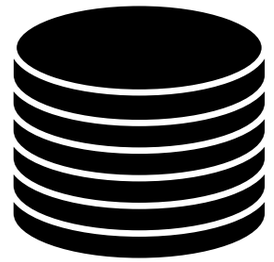
Simplified view on today's web development



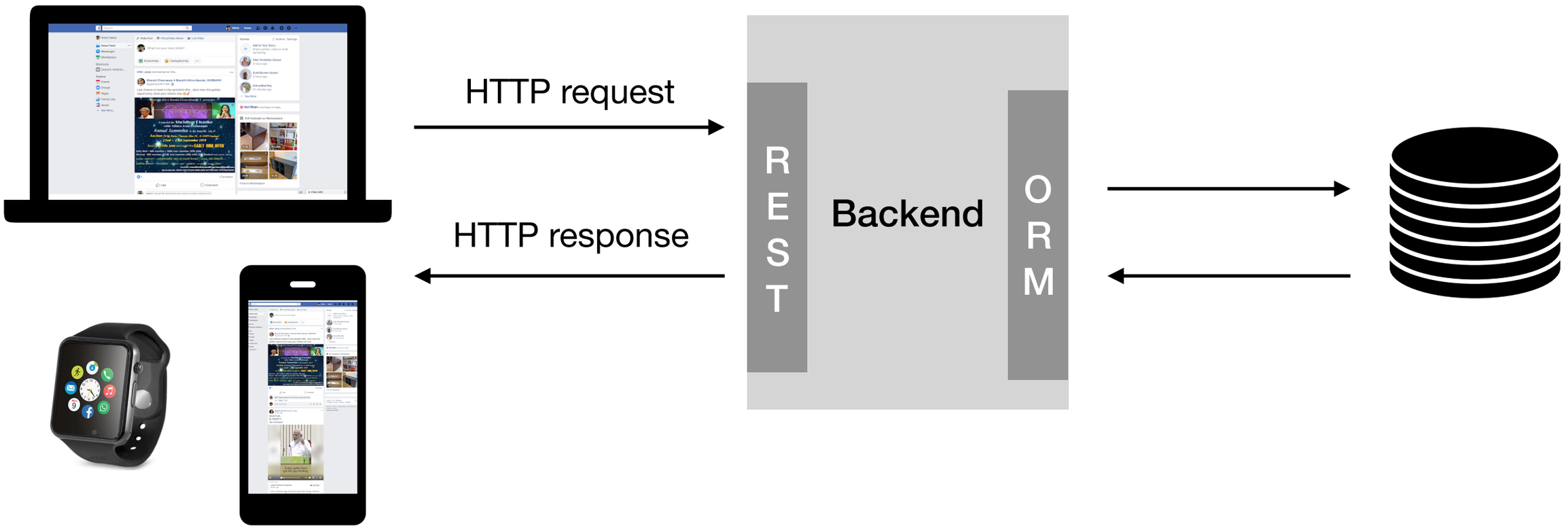
HTTP request

HTTP response

Backend



Simplified view on today's web development



Frontend: Key responsibilities and important concepts to get familiar with

- Frontend is what your users see- can be accessed on mobile as an app, or as a mobile website, or on laptops etc.

- Accepts request from users and forwards it to the backend, subsequently receives response from backend and shows it to the user

Important concepts-

- HTML, CSS, Javascript, Typescript
- HTTP requests
- Ajax (or asynchronous calls to backend)
- Javascript frameworks and libraries (e.g. Angular, React, Vue.js)
- Styling frameworks (e.g. Bootstrap, Material)
- Package manager (e.g. NPM)
- Single Page applications (SPA)
- Developer tool in the browser
- Responsive design
- Browser compatibility
- …

Frontend: Key responsibilities and important concepts to get familiar with

- Frontend is what your users see- can be accessed on mobile as an app, or as a mobile website, or on laptops etc.

- Accepts request from users and forwards it to the backend, subsequently receives response from backend and shows it to the user

Important concepts-

- HTML, CSS, Javascript, Typescript
- HTTP requests
- Ajax (or asynchronous calls to backend)
- Javascript frameworks and libraries (e.g. Angular, React, Vue.js)
- Styling frameworks (e.g. Bootstrap, Material)
- Package manager (e.g. NPM)
- Single Page applications (SPA)
- Developer tool in the browser
- Responsive design
- Browser compatibility
- …

Backend: Key responsibilities and important concepts to get familiar with

- Backend handles business logic for your application- you typically write single backed application which is then consumed by various frontends
- Coordinates with frontends and the database

Important concepts-

- REST API
- JSON data format
- Programming languages (e.g. PHP, Python, Java)
- Frameworks (e.g. Django, .NET core, Rails)
- Object Relational Mapping (ORM)
- HTTP Request handling and routing
- Session management
- Security concerns
- …

Backend: Key responsibilities and important concepts to get familiar with

- Backend handles business logic for your application- you typically write single backed application which is then consumed by various frontends
- Coordinates with frontends and the database

Important concepts-

- REST API
- JSON data format
- Programming languages (e.g. PHP, Python, Java)
- Frameworks (e.g. Django, .NET core, Rails)
- Object Relational Mapping (ORM)
- HTTP Request handling and routing
- Session management
- Security concerns
- …

Database: Key responsibilities and important concepts to get familiar with

- Database stores user and application data
- With the availability of ORMs, developers deal less with database directly

Important concepts-

- Types of databases (e.g. Relational, non-relational)
- Database venders (e.g. MySQL, PostgreSQL, MongoDB)
- Database backups
- Database query languages (e.g. SQL)
- Database scripting languages (e.g. PL/SQL)
- …

Database: Key responsibilities and important concepts to get familiar with

- Database stores user and application data
- With the availability of ORMs, developers deal less with database directly

Important concepts-

- Types of databases (e.g. Relational, non-relational)
- Database venders (e.g. MySQL, PostgreSQL, MongoDB)
- Database backups
- Database query languages (e.g. SQL)
- Database scripting languages (e.g. PL/SQL)
- …

Frameworks

- One can do web development without frameworks, but using them simplifies many routine tasks and speeds up the development, also makes sure applications are scalable

- Choosing a correct framework depends on multiple factors such as programming language preference, framework documentation, community support etc.

- There are different frameworks for frontend development, backend development, and also for styling

Frontend (javascript) Frameworks

- Enforces developers to write HTML, CSS and javascript in a certain way to make the application easier to develop and maintain
- Easy handling of Ajax calls, HTTP requests
- Few popular frontend frameworks and libraries:

Styling Frameworks

- Provides ready to use css classes, components and grid system based on proven design standards, so that developers do not have to write css from scratch

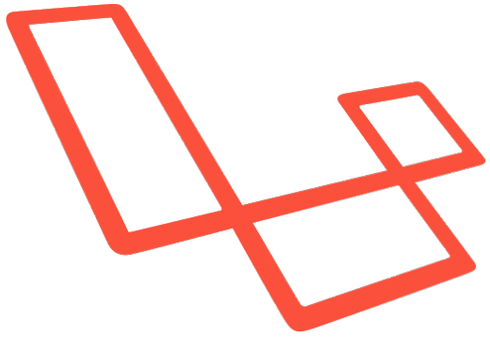
- Few popular styling frameworks:

Backend Frameworks

- They provide tools and libraries that simplify common web development tasks, including routing URLs to appropriate handlers, interacting with databases through ORM, supporting sessions and user authorisation, formatting output (e.g. HTML, JSON, XML), and improving security against web attacks.

- There also exist API query languages such as GraphQL to describe data in your API

- Few popular backend frameworks:



Project Scaffolding

- Helps to quickly set up a project skeleton (hopefully) with best practices
- One can scaffold a project by using CLI tools such as angular cli for angular projects, or by cloning a repository from GitHub, GitLab etc.
- Few popular scaffolding tools:

Slush

Project management tools

- Helps to organise development work within teams, manage user stories, deadlines, generate project analytics and many more tasksOne can scaffold a project by using CLI tools such as angular cli for angular projects, or by cloning a repository from GitHub, GitLab etc.

- Few popular project management tools:

Communication tools

- Helps to communicate with other team members through dedicated channels, direct messages and rich content such as video, images, Gif etc.

- Provide search functionality and integrations to other applications such as confluence, gitlab etc.

- Example:

Final words..

As a web developer you cannot live without..