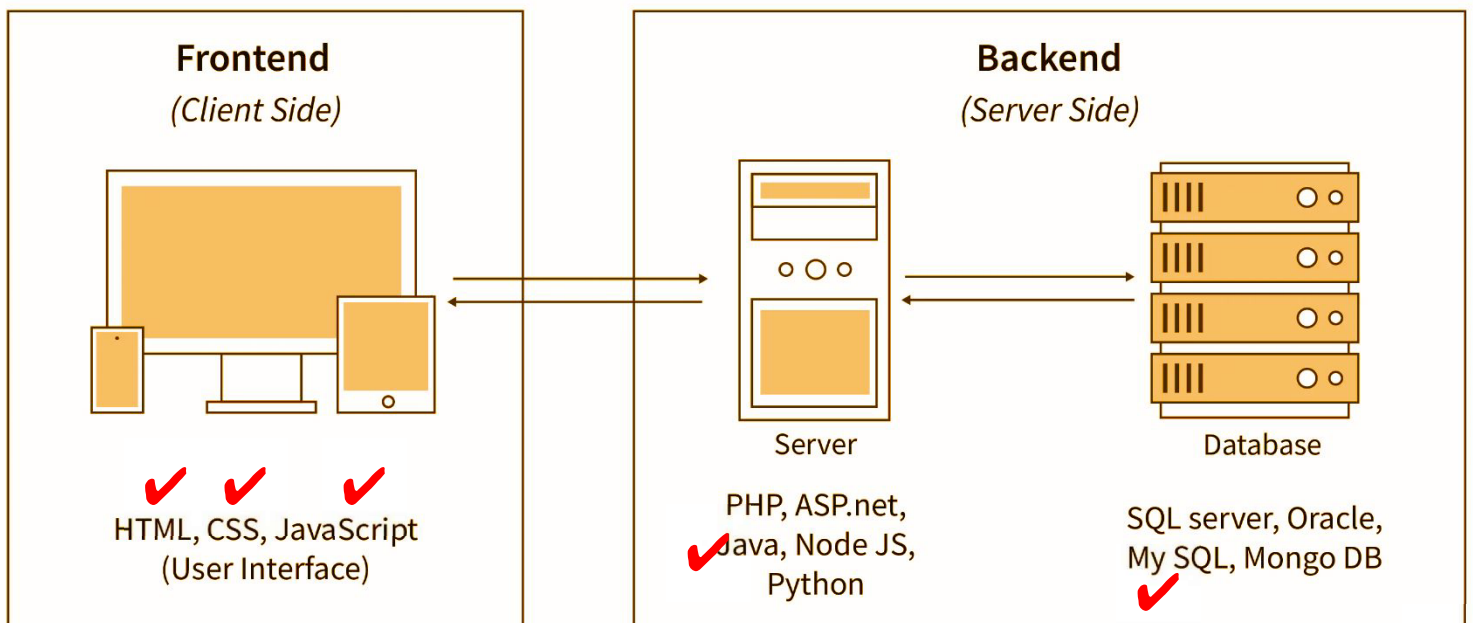


Full-Stack Development

Dr Harshad Prajapati
15 Aug 2023, 28 Nov 2023

Full-Stack Web Development



Web is Required for Digital Existence

- Physical Products:
 - Notebook, Pen, Cars, Grocery items, bakery items
- Digital Products:
 - Software
 - Movies
 - e-books
- Physical products and physical services have digital existence:
 - Online sell/store
 - Online doctor consultation
- Digital existence started with the rise of web.

3

Rise of Web

- The sophisticated apps on big screen devices and small screen devices have become possible due to continuous enhancements of web:
 - Starting from HTML, CSS, VBScript, JavaScript to Standard Based Technologies.
 - Difference Devices (Screen sizes) with different resolutions:
 - CSS Media Queries for desktop, laptop, tablet, and smartphones.
 - Increased network connectivity and bandwidth.

4

Rise of Web

- **Native app** like capability:
 - ActiveX, Java applets, Flash.
 - XMLHttpRequest API (XHR) to JavaScript - AJAX.
 - Rise of Single Page Application (SPA).
 - New frameworks, Fetch API, Advanced JavaScript.
- **Open source** based **browsers**:
 - Earlier, different browsers supported different things.
 - Need to detect device and should have code supporting that device/browser.

5

Rise of Web

- Improvement in content delivered from **backend**:
 - **Statically served pages** from server.
 - **Scripting** to generate **dynamic content** on server.
 - **Framework** based development (**MVC**).
 - Implementation in backend to **support AJAX**.
 - Separation of content from its representation:
 - **API** and its **reusability** in Web App and Mobile App.
 - Support for **Single Page Application**.
 - Rise of **Declarative** style of **programming** and support in frameworks.
 - Security Protocols: **OAuth** and **OIDC**.

6

Rise of Web

- Mobile Web:
 - Web applications are replacing many desktop applications.
 - For mobile platforms, have built-in app distribution mechanism (Example: Play Store).

7

Modern App Development

8

Requirements of Modern Applications

- **Multi-factor authentication:** Username and password, PAN and OTP, fingerprint, etc.
- **Third party authentication:** Google/Facebook/GitHub Login (SSO)
- **Payment Gateway Integration:** Support of payment with UPI, Netbanking, Credit card, etc.
- **Email/SMS Notification:** Notify users about important events via various means.
- **Search Engine Optimization:** Improve the Google Search ranking of app.

9

Web Apps vs Web Sites

- **Website:**
 - Websites were generally **content based**: providing **just information** and very **little user interactivity**.
- **Web apps:**
 - They provide **user interactivity** (input/output)
- **Distinction** between **Websites** and **web apps** have **blurred**:
 - **Frameworks** such as **Angular** and **Backbone** are examples of these types of frameworks.

10

Roles in App Development

- Business Owner
- Product Manager
- Designers
- Backend
- Frontend
- Quality Assurance (QA)
- DevOps (Development + Operations)

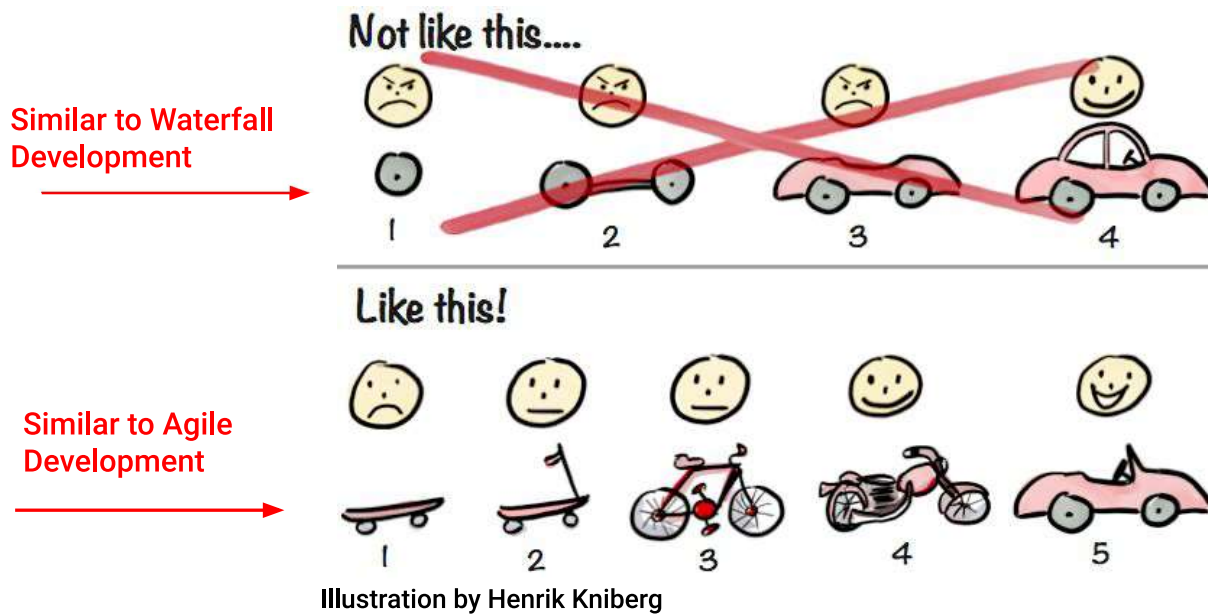
11

Development Methodologies

- Waterwall (Traditional):
 - Requirements, Analysis, Design, Coding, Testing, Maintenance.
- Agile (Contemporary):
 - Incremental development and delivery.
 - Scrum and Kanban frameworks.

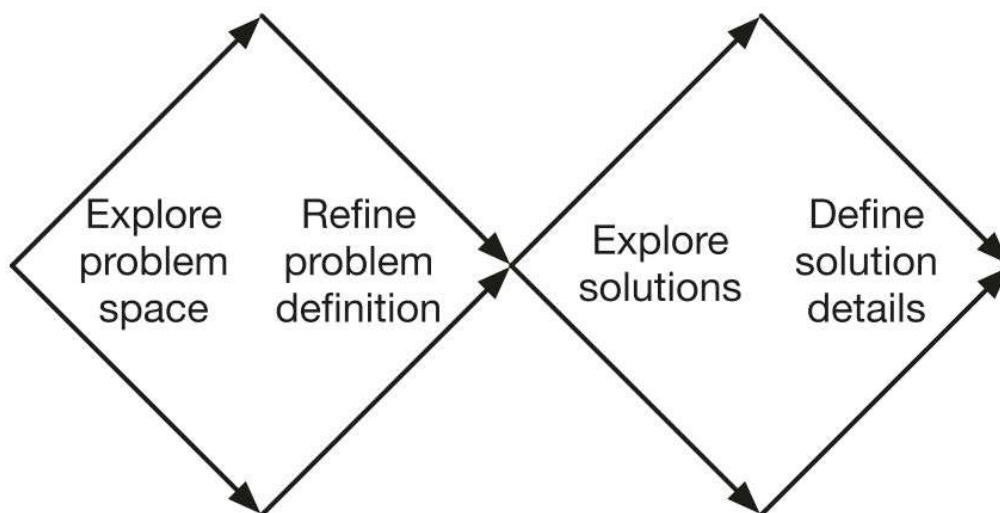
12

Minimum Viable Product (MVP)



13

Double-diamond Model of Problem-Solution



A developer spends around only 30% time in coding; remaining in thinking, exploring, experimenting.

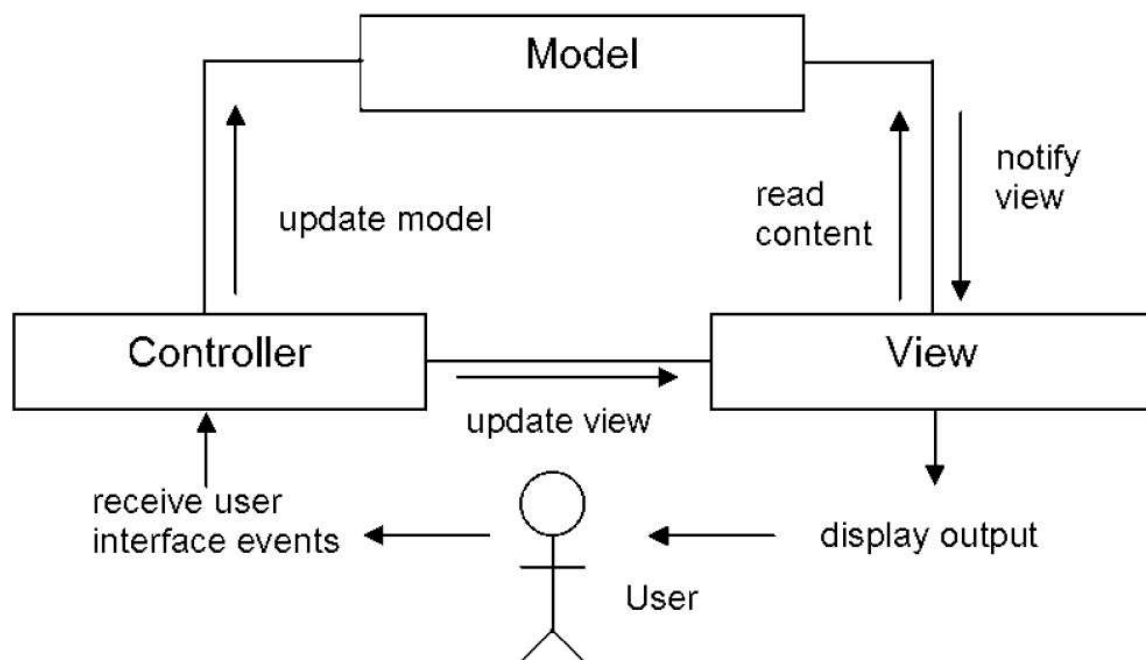
Image Source: Chapter 2, Planning Your Work, The Full Stack Developer, Chris Northwood

14

Application Architecture

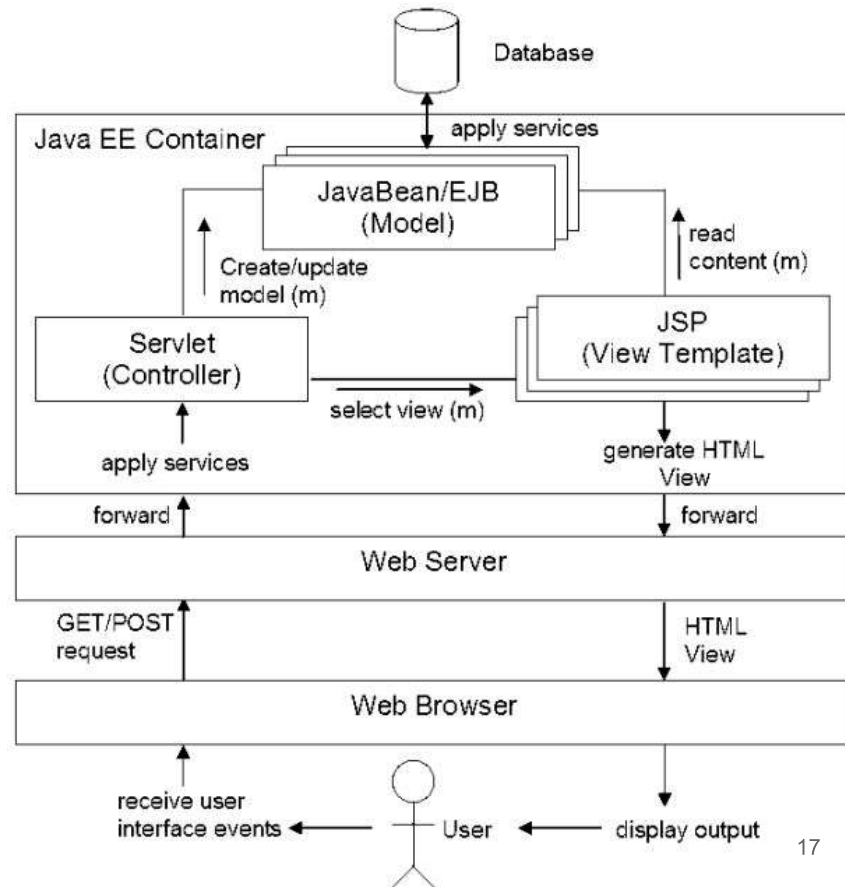
15

Model-View-Controller (MVC) Architecture



16

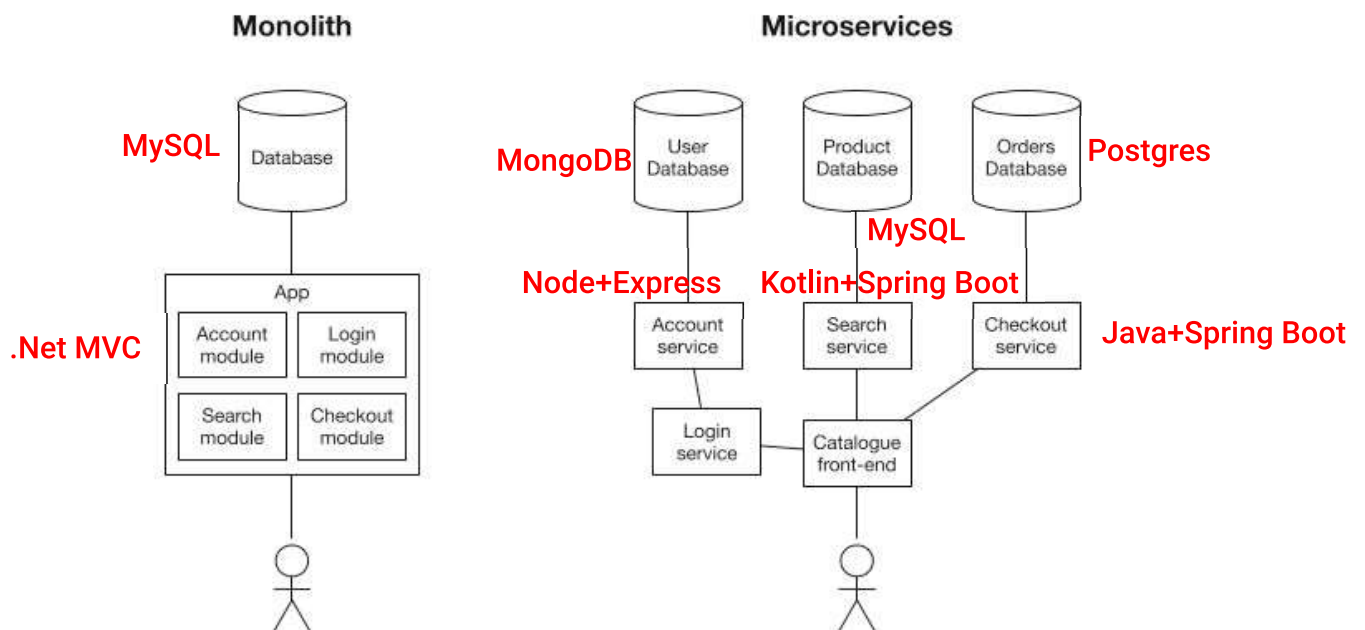
MVC Architecture in JEE



17

Image Source: High Quality Web-Application Development on Java EE Platform, April 2009, DOI: 10.1109/IADCC.2009.4809267

Traditional versus Modern App Architecture



18

Microservice

- ✓ A **microservice** is a **small service** that takes **responsibility** for **one part** of a **system**.
- ✓ Microservice can be **developed**, **deployed**, and **scaled independently** of any other part of the system.

19

Backend Development

- Backend development generally includes:
 - **Build Automation Tool**: Maven, Gradle.
 - **Database**: SQL, NO-SQL.
 - **Authentication**: OIDC protocol.
 - **Authorization**: OAuth2 protocol.
 - ✓ ○ **Access Token** and **Refresh Token**.
 - Access token to **access API**.
 - Refresh token is to **refresh** the **expired access token**.

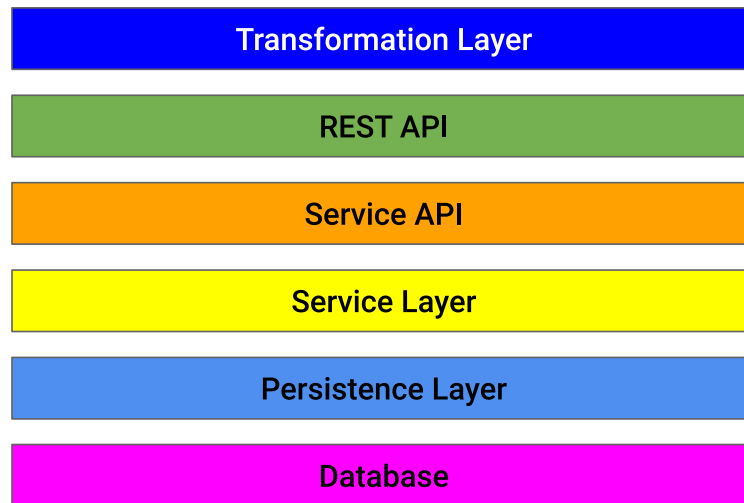
--> OpenID Connect (OIDC) is an identity protocol with a standardized format for exchanging authentication information between websites and authentication services.

OIDC leverages OAuth 2.0 and adds an identity layer to the authorization framework. The new layer enables websites and applications to authenticate users through other service providers

--> OAuth 2 focuses on authorizing access via an authentication server (like Facebook) to user resources to a client on behalf of its owner.

Using OAuth 2.0, resource owners can authorize third-party applications to access their personal data, such as contacts or profile information, through authentication and resource servers – without sharing their login credentials

Backend Development Stack



21

Frontend Development

- Markup:
 - Headings, Hyperlinks, Images, Form, Videos.
- Document Object Model (DOM).
- DOM Manipulation.
- CSS box model: content, padding, border, and margin.
- Layout: CSS flexbox, CSS Grid, CSS Media Queries.

22

HTML

- HTML standard specifies:
 - Tags.
 - Document Object Model (DOM) (manipulating using JavaScript).
 - CSS.

23

JavaScript

- JavaScript was **originally** designed at **Nescape** to **manipulate DOM using API**.
 - **Word Java** was used due to **rising popularity** of **Java** at that time.
- **Microsoft** added its **own variant**, **JScript**.
 - Added new feature **XMLHttpRequest** (XHR for short)
- The **European Computer Manufacturers Association (ECMA)** decided to attempt to **merge competing implementations** into a **new standard**, **ECMAScript**.
 - ES6 (ECMAScript 2015)
 - ES2018 (ECMAScript 2018)
- **Implemented** by many **browsers**: **Chrome, Firefox, Edge, Safari, Opera**, etc.

24

Contract between Frontend and Backend

- JSON
 - JavaScript Object Notation.
 - Easy to read by humans.
 - Easy for browser to interpret.
 - Easy to process by JavaScript.
- XML
 - Still used by enterprise applications/systems.
 - Used in legacy applications/systems.

25

Frontend Frameworks

- jQuery
- Backbone
- Ampersand
- React (is a UI library)
- Angular
- Meteor
- Ember
- Vue.js
- Next.js (React based Full stack framework)
- Nuxt.js (Vue based Full stack framework)

26

Design Principles (Applicable to Frontend and Backend)

- ✓ ● SOLID Principles:
 - Single Responsibility Principle.
 - Open/Closed Principle.
 - Liskov Substitution Principle.
 - Interface Segregation Principle.
 - Dependency Inversion Principle.
- DRY
 - Don't repeat yourself.

Further Reading: Clean Code, Robert C. Martin

27

Frontend Design Principles

- Responsive Design
- Mobile First

28

User-Centered Design

- **Sketching:** **freehand** sketch made with a pencil or pen.
- **Wireframes:** Detailed black and white **layout** of the website page, focus on **placement** of **elements**.
- **Prototype:** **Interactive** version of the **wireframe**. (Black and White)
- **Mockup:** **Beautiful** version of **wireframes**. (Colors, images, typography, theme)

29

User-Centered Design

- **User stories:**
 - Concise and simple descriptions of a **feature** told from the **perspective** of the end **user**.
 - They express **requirements** and **functionalities** of a software system in a **user-centered manner**.
- **User Interface (UI) and User Experience (UX):**
 - UI and UX are not same.
 - UI: **Visual elements**. How a product looks and interacts visually.
 - UX: **Overall user experience** and **satisfaction**.
 - Alexa has no visible user interface, but has a great user experience.

30

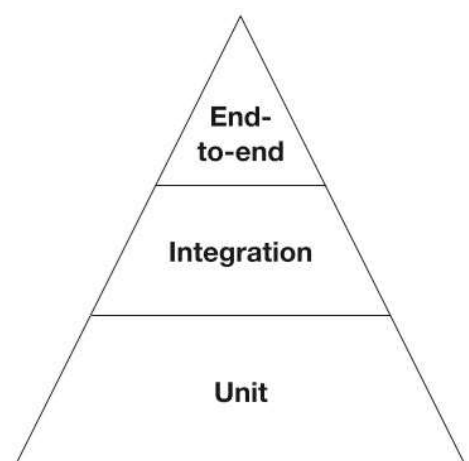
Technologies in Fullstack Development

- Frontend
 - HTML, CSS, JavaScript
 - TypeScript
 - React, Angular, Vue
- Backend
 - Django, Ruby on Rails, CakePHP, Laravel, Spring Boot, Flask, Koa, Phoenix, Asp.NET
- Fullstack
 - HTML, CSS, JavaScript, JQuery, PHP, Ajax, MySQL, Linux, UNIX, and C++/Java/Python. Apache NGINX

31

Testing our Application

- Different Types of Testing
 - Unit Testing
 - Integration Testing
 - End-to-end Testing (e2e)
 - System Testing
 - Acceptance Testing
 - Regression Testing
- Ways of Testing
 - Manual
 - Automated (Mocha, Selenium, Cypress)



Testing pyramid

32

Development with Testing

- Test-Driven Development (TDD)
 - Red-green-refactor
- Behavior-Driven Development (BDD)

33

IDEs

- IntelliJ Idea
- VS Code
- Netbeans
- Eclipse
- Spring Tool Suite
- Sublime

34

Web based IDEs and Playgrounds

- GitHub's Codespace
 - <https://github.com/features/codespaces>
- <https://codepen.io/>
- <https://replit.com/>

35

Publish/Deploy App

- DNS registration (GoDaddy, Hostinger, etc.).
- Deploy app on deployment platforms.

36

Free Deployment Platforms

- Railway
- Netlify
- GitHub Pages
- Vercel
- Google Site

37

CI/CD Pipeline

- Continuous Integration:
 - Versioning, Merge/Pull Request.
 - GitHub, GitLab,
- Continuous Delivery:
 - Continuous Deployment.
 - Jenkins, GitHub Actions, Travis CI, GitLab CI/CD, many more.

38

Maintaining and Improving App

- Maintaining
 - Backups
 - Replication: Master-slave
 - Disaster Recovery
- Improving
 - Marketing
 - Sales
 - R&D
 - Customer Service
- Scaling: auto scaling

39

Maintaining and Improving App

- Bug fixing
- Refactoring and rewriting

40

Development Tips

- Choosing **right** programming languages or **Frameworks**.
- Follow **coding** style **guidelines**.
- Code **reviews**.
- Define **CI/CD pipeline** from the **beginning**.
- Don't underestimate the **importance** of **required infrastructure**.

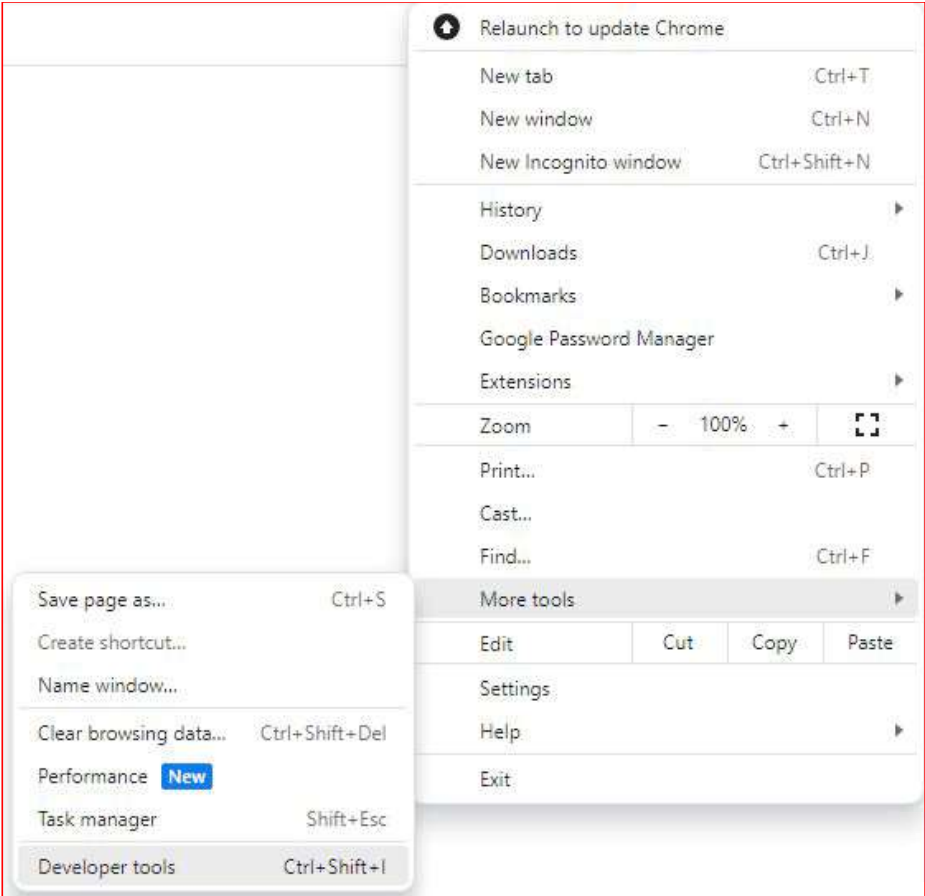
41

Developer Tools

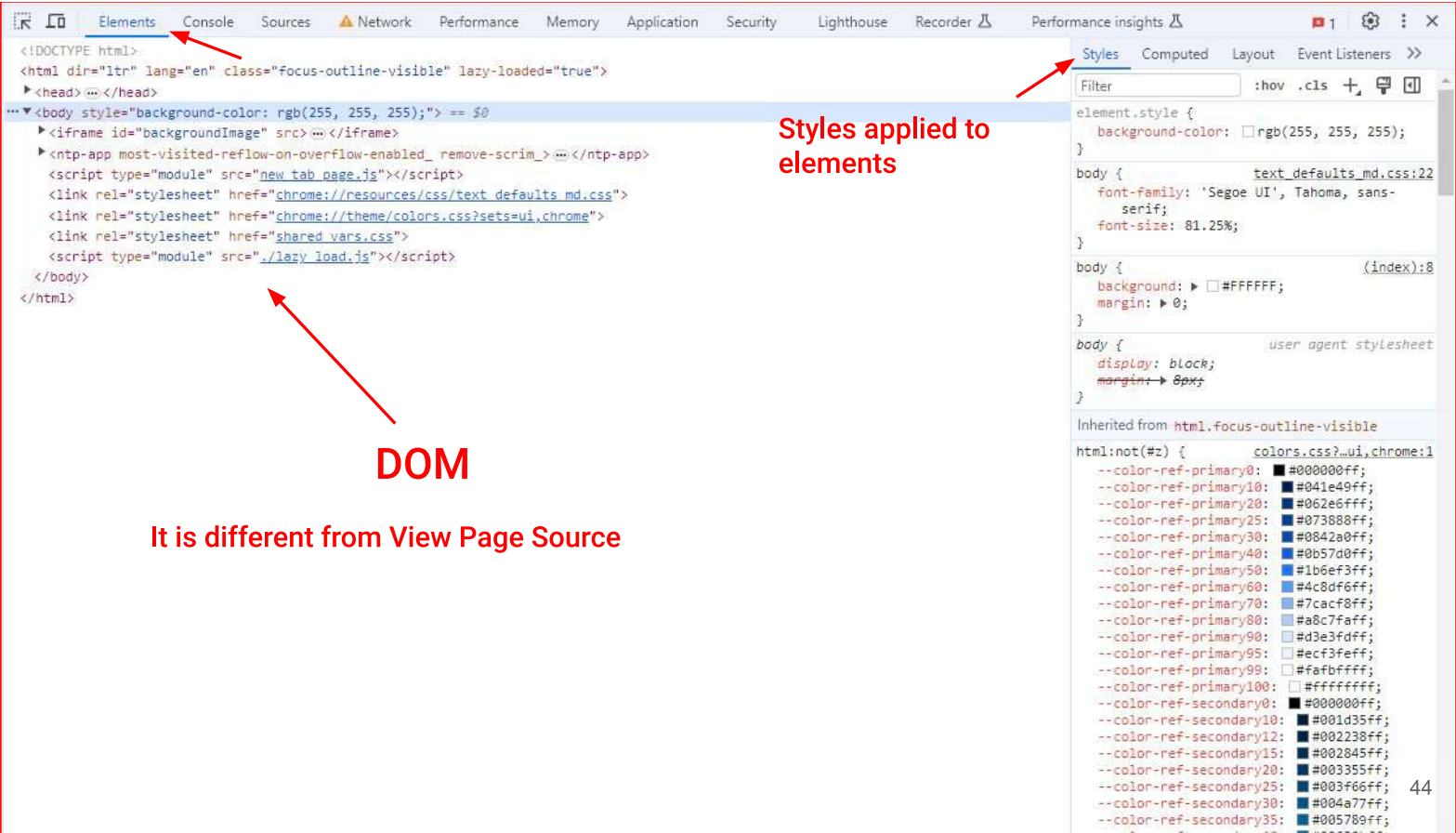
42

Developer Tools

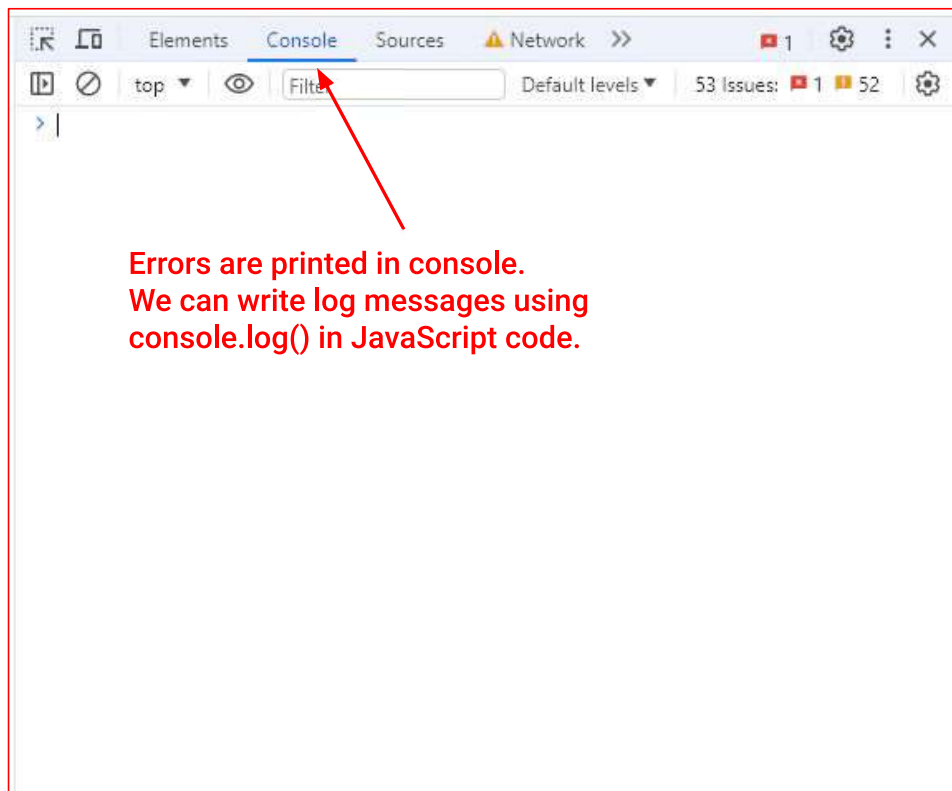
In Chrome browser



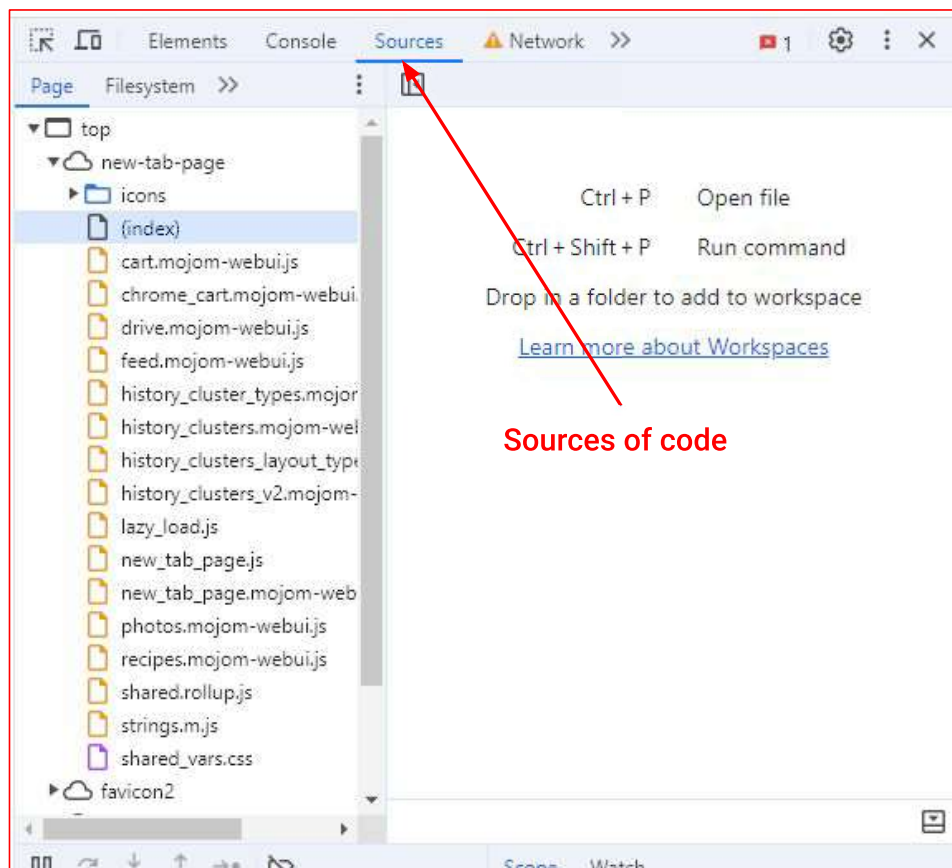
43



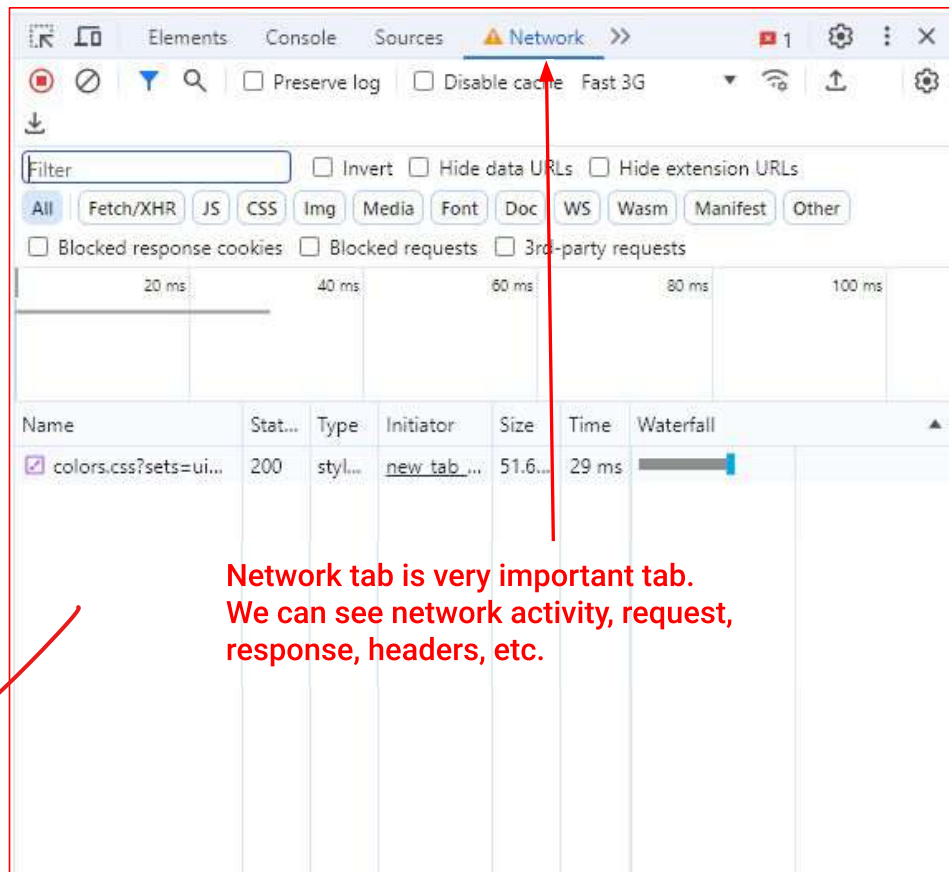
44



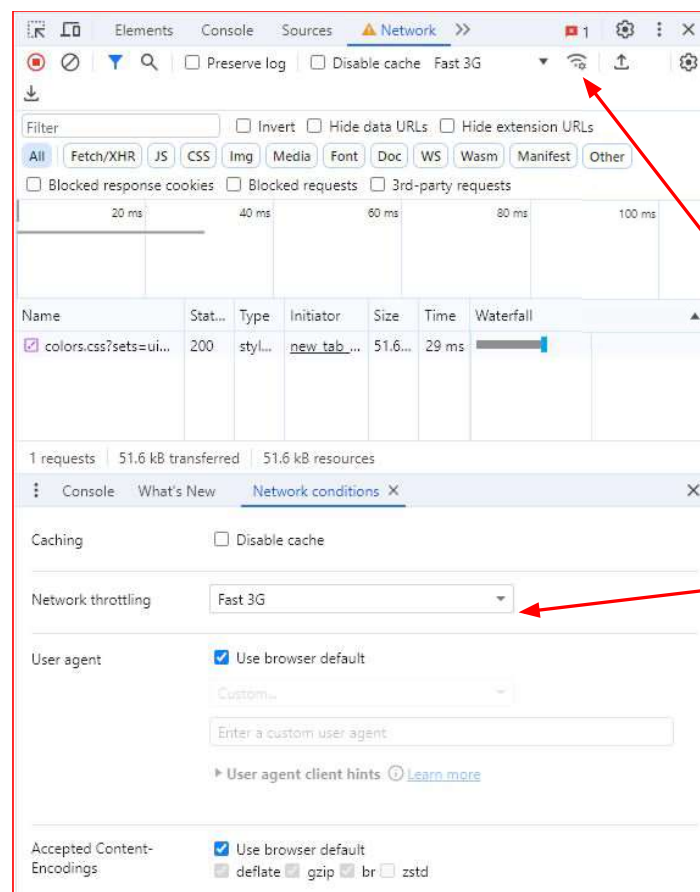
45



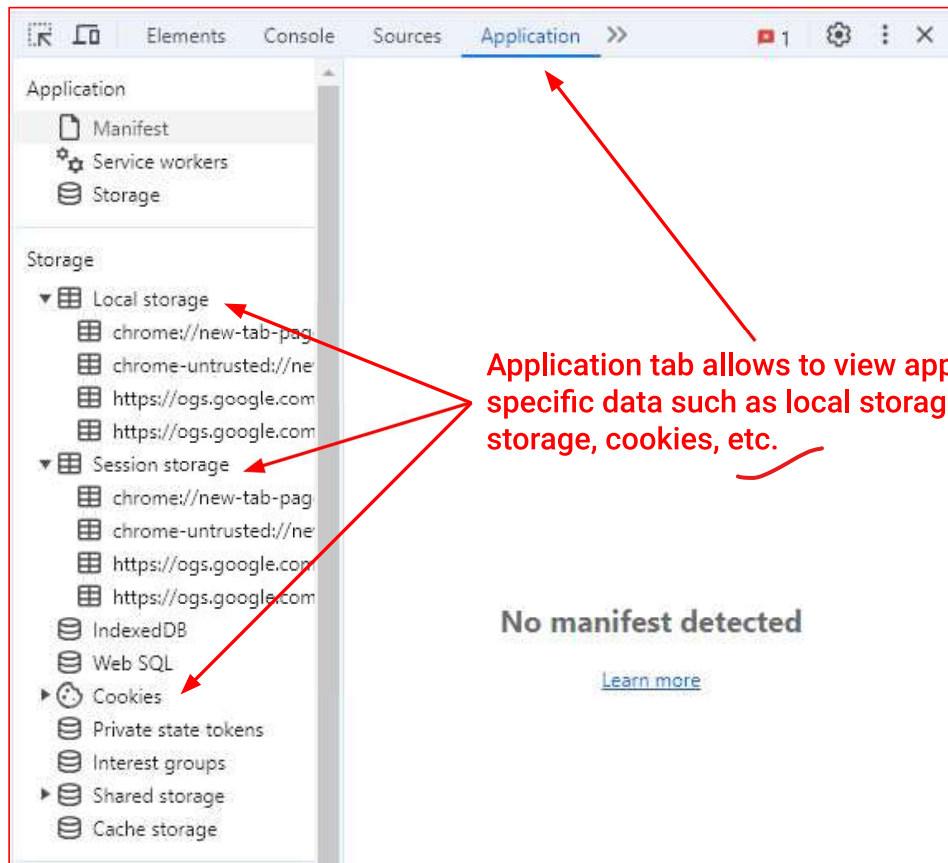
46



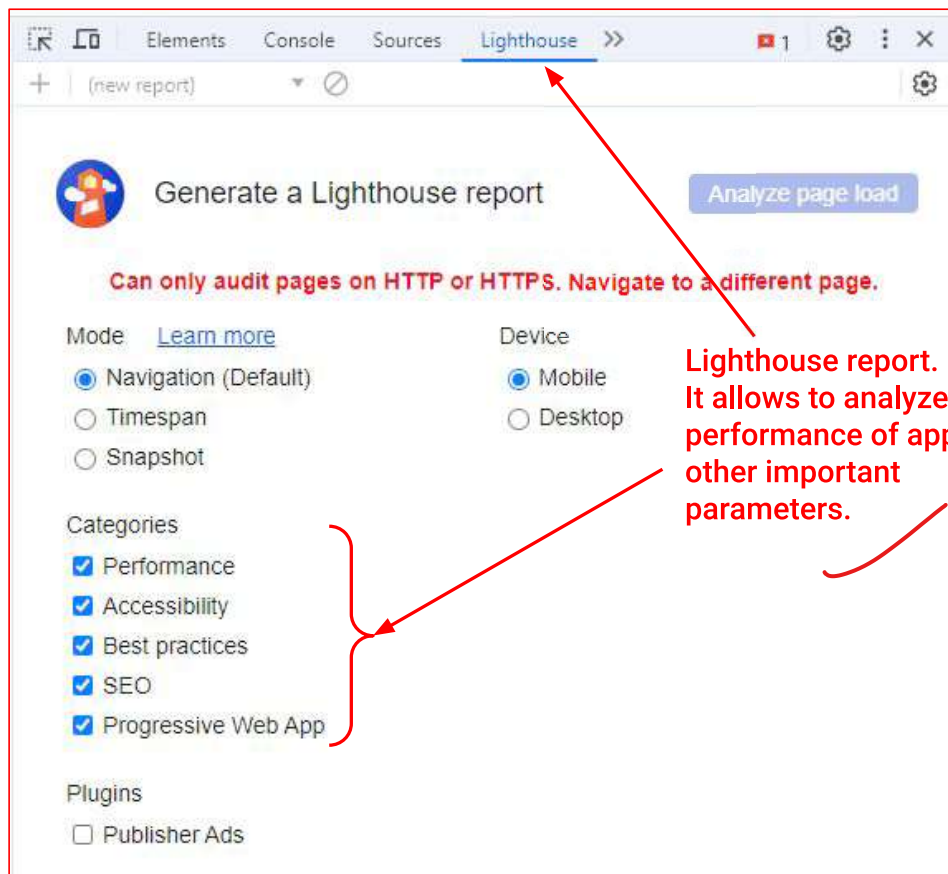
47



48



49



50

What to Learn?

- Frontend:
 - Foundation of Web (HTML, CSS, JavaScript).
 - Advanceds features of JavaScript.
 - React Library.
 - React Router.
 - Redux state management.
- Backend:
 - Hibernate.
 - Spring MVC.
 - Spring Boot.
 - Spring Security.

51

References

- Software Development From A to Z, A Deep Dive into all the Roles Involved in the Creation of Software, Olga Filipova and Rui Vilao, APress

52