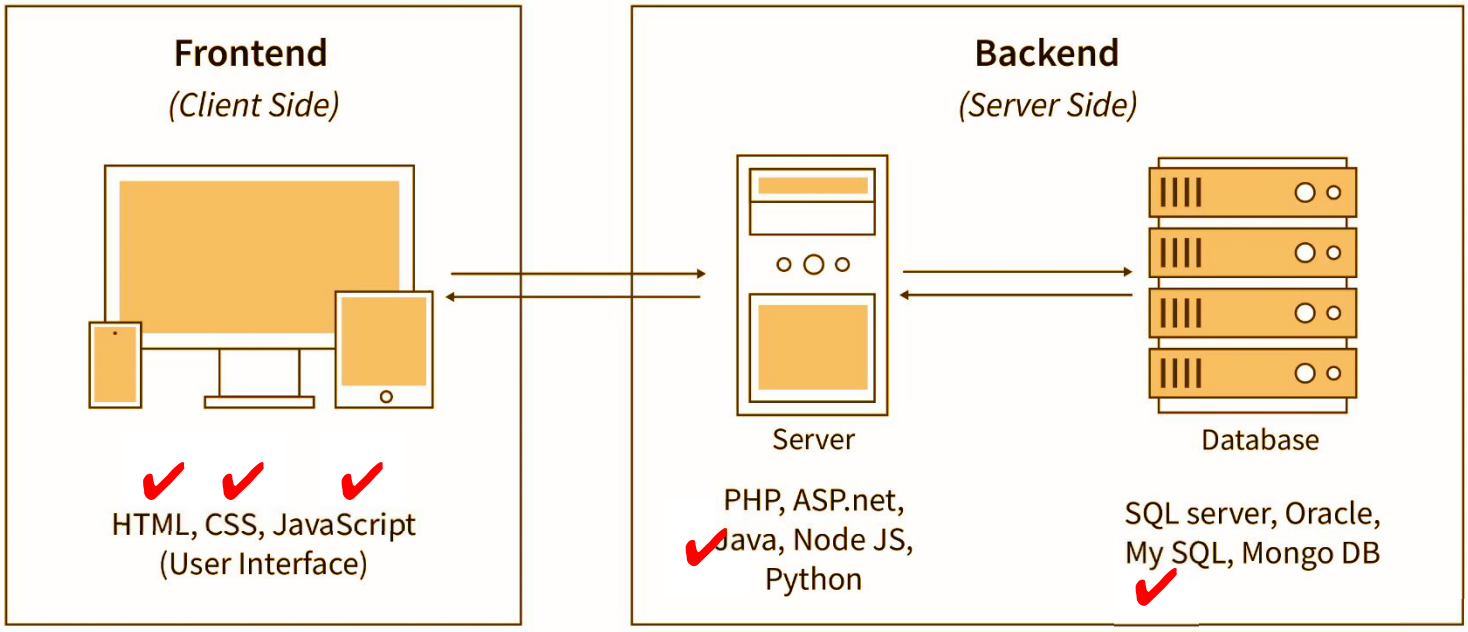
Full-Stack Development

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.

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.

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.

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.

Rise of Web

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

Modern App Development

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.

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.

Roles in App Development

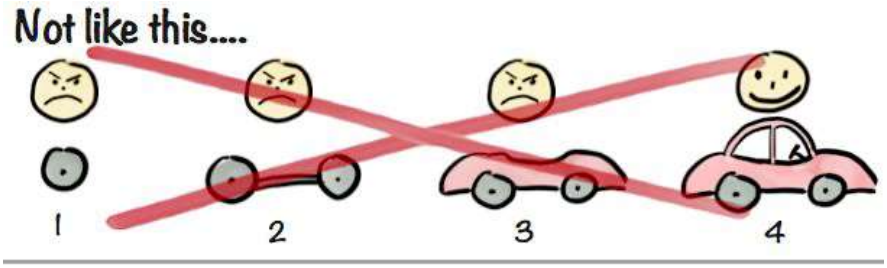
- Business Owner
- Product Manager
- Designers
- Backend
- Frontend
- Quality Assurance (QA)
- DevOps (Development + Operations)

Development Methodologies

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

Minimum Viable Product (MVP)

Similar to Waterfall Development



Similar to Agile Development

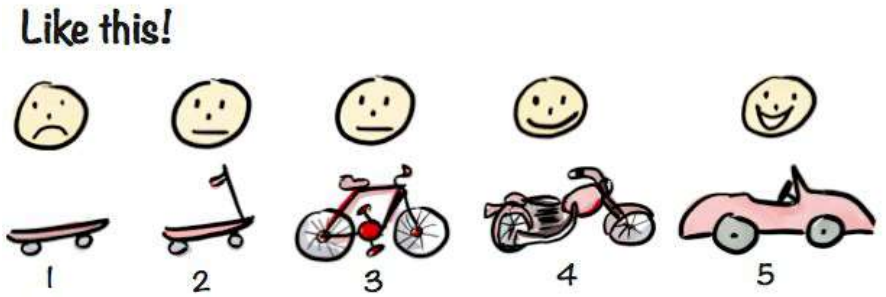
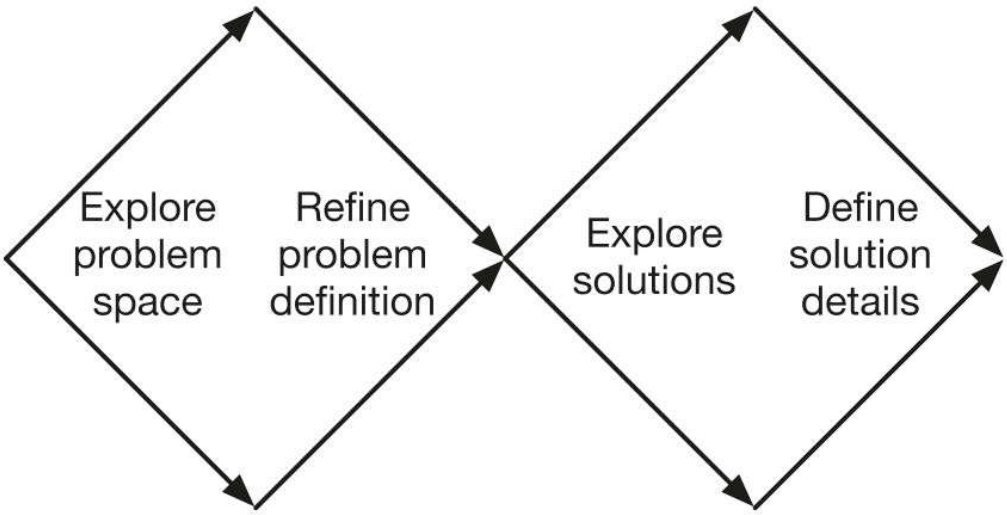


Illustration by Henrik Kniberg

Image Source: https://blog.crisp.se/2016/01/25/henrikkniberg/making-sense-of-mvp

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

Application Architecture

Model-View-Controller (MVC) Architecture

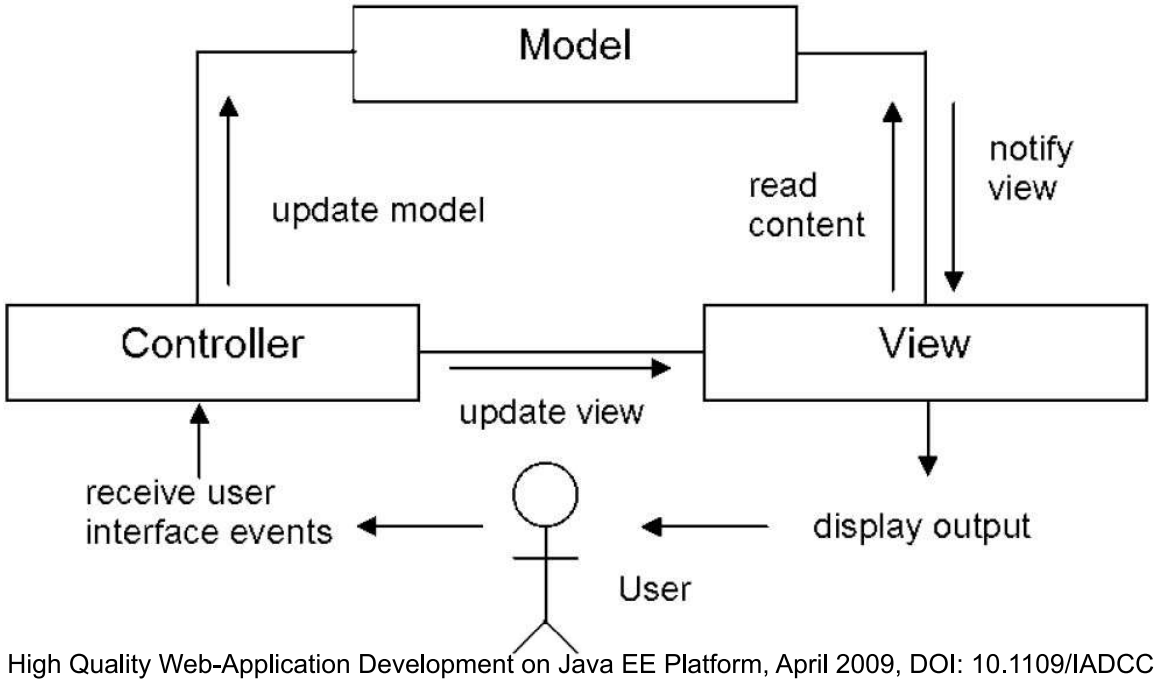


Image Source: High Quality Web-Application Development on Java EE Platform, April 2009, DOI: 10.1109/IADCC.2009.4809267

MVC Architecture in JEE

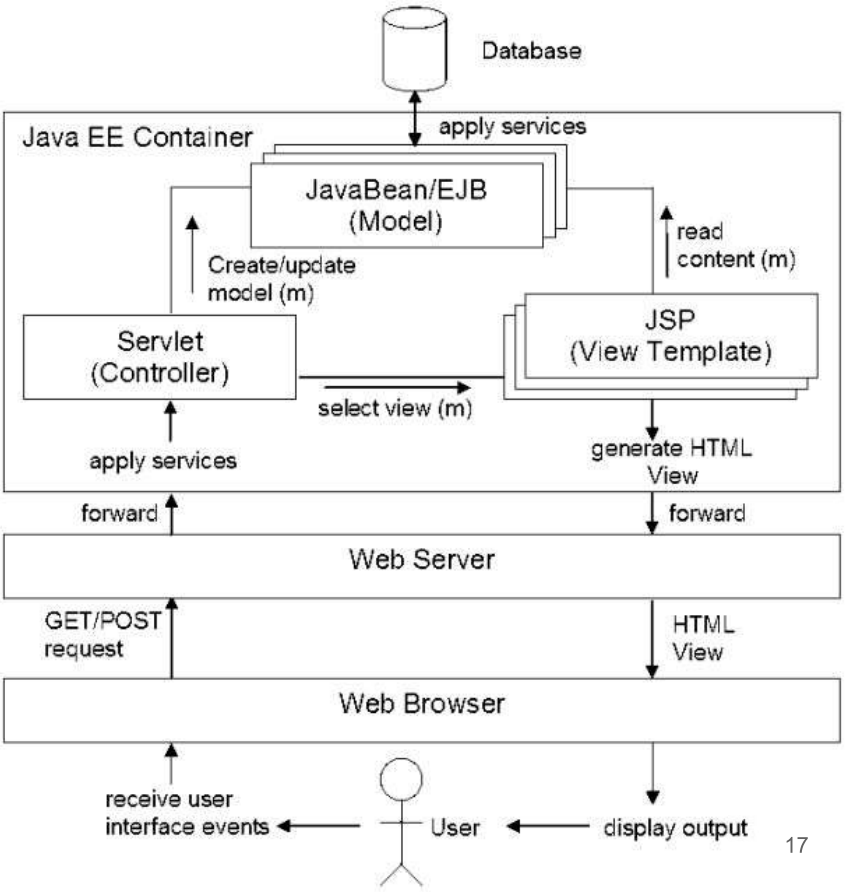
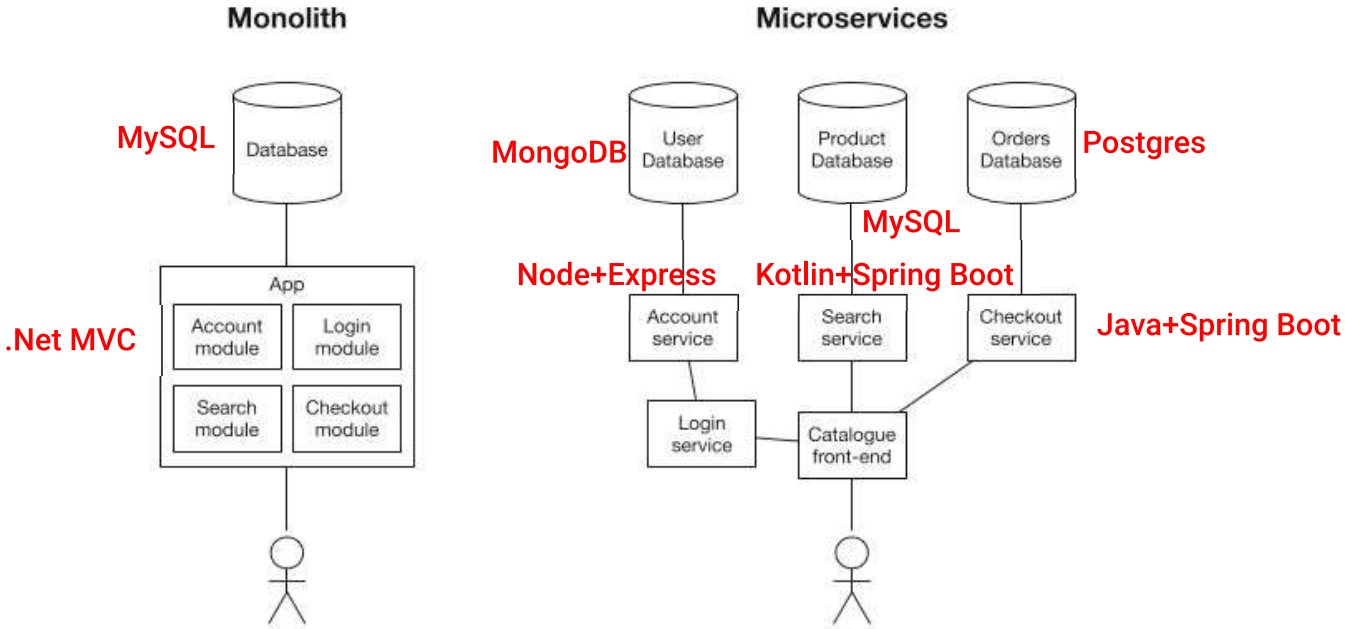


Image Source: High Quality Web-Application Development on Java EE Platform, April 2009, DOI: 10.1109/IADCC.2009.4809267

Traditional versus Modern App Architecture



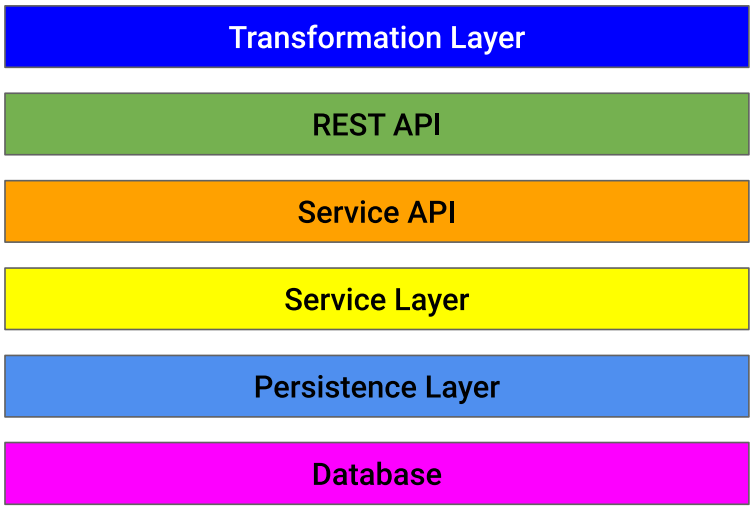
Microservice

- 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.

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.

Backend Development Stack



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.

HTML

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

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.

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.

Frontend Frameworks

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

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

Frontend Design Principles

- Responsive Design
- Mobile First

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)

User-Centered Design

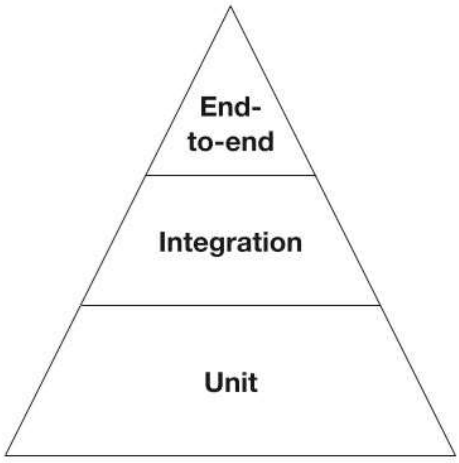
- User stories:
  - Concise and simple descriptions of a feature told from the perspective of the end user.
  - They express requirements and functionalities of a software system in a user-centered manner.
- User Interface (UI) and User Experience (UX):
  - UI and UX are not same.
    - UI: Visual elements. 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.

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

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

Development with Testing

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

IDEs

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

Web based IDEs and Playgrounds

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

Publish/Deploy App

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

Free Deployment Platforms

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

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.

Maintaining and Improving App

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

Maintaining and Improving App

- Bug fixing
- Refactoring and rewriting

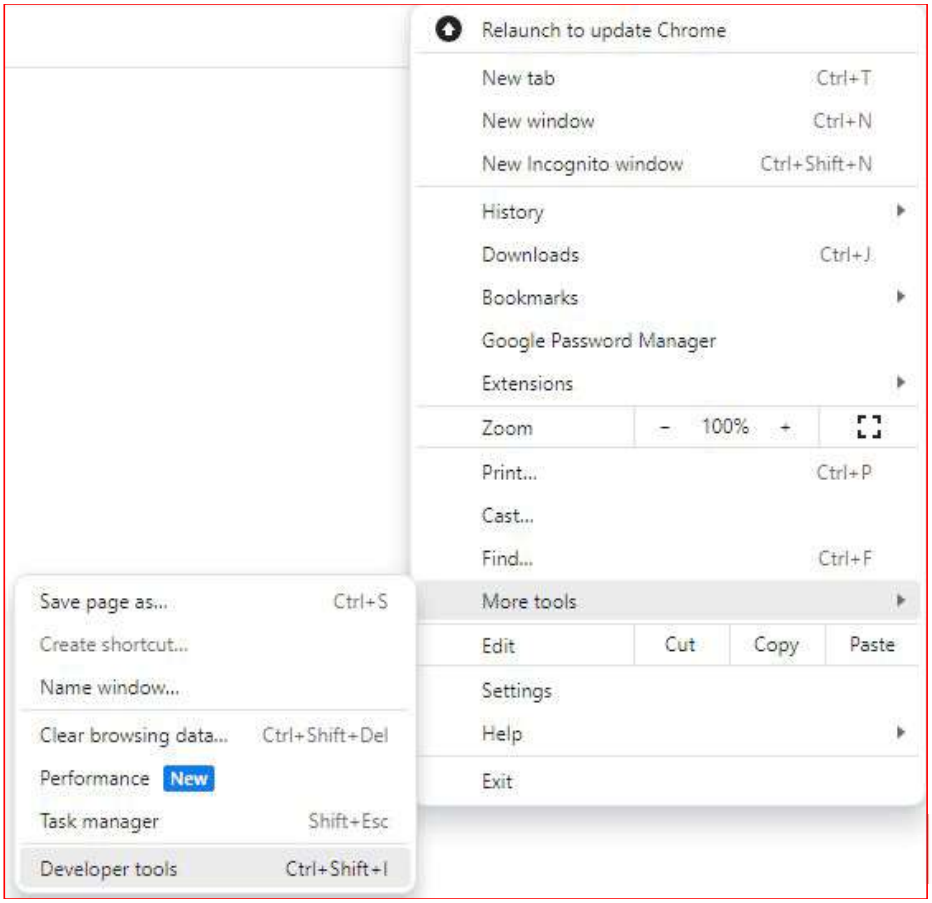
Development Tips

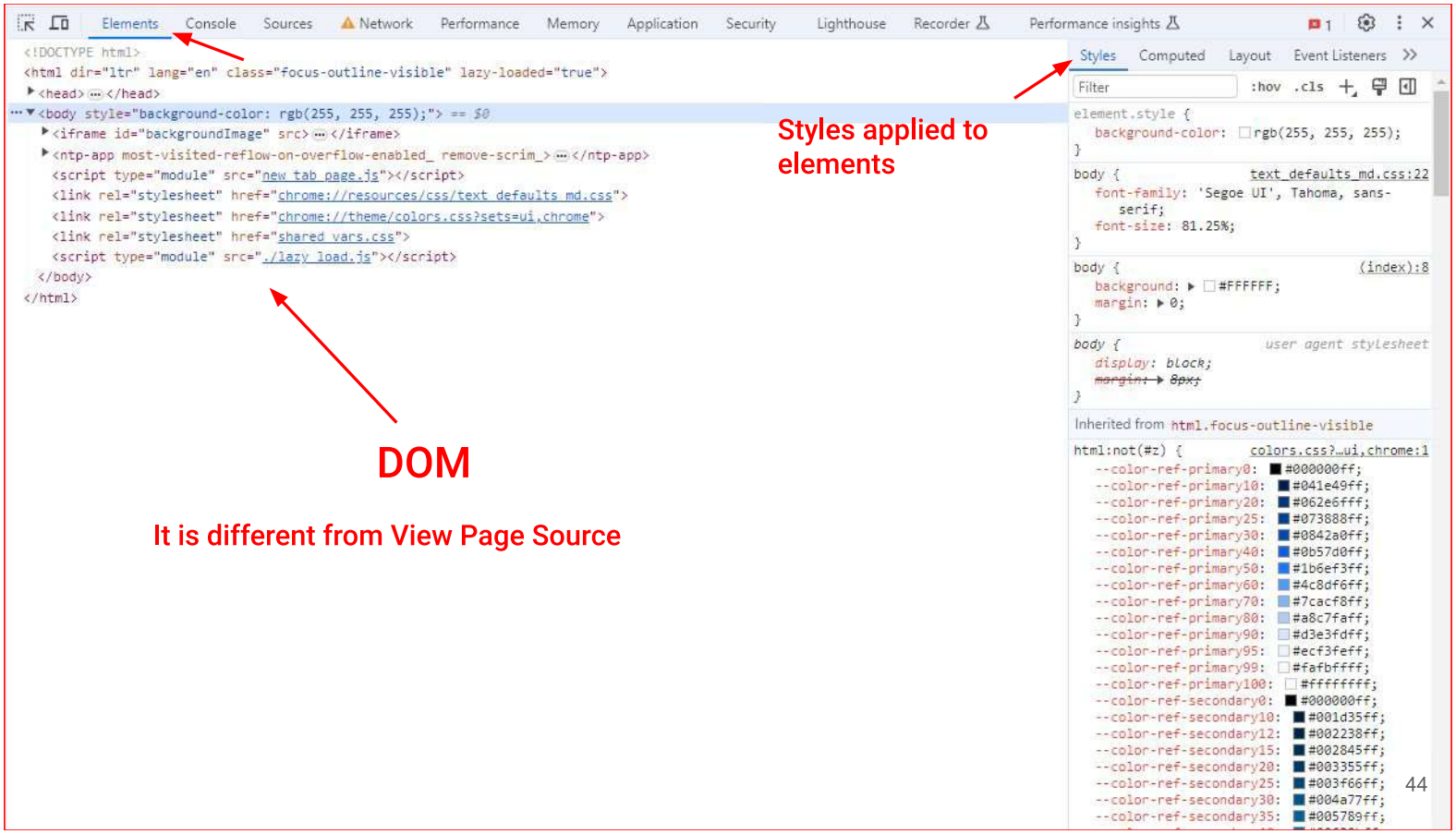
- Choosing right programming languages or Frameworks.
- Follow coding style guidelines.
- Code reviews.
- Define CI/CD pipeline from the beginning.
- Don’t underestimate the importance of required infrastructure.

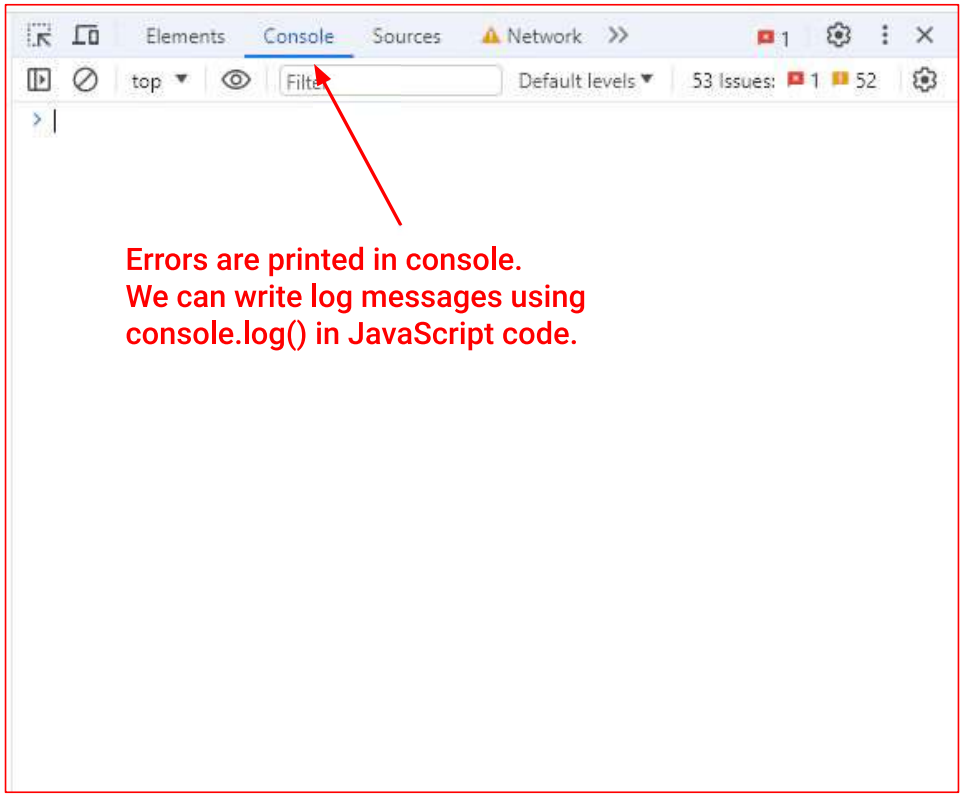
Developer Tools

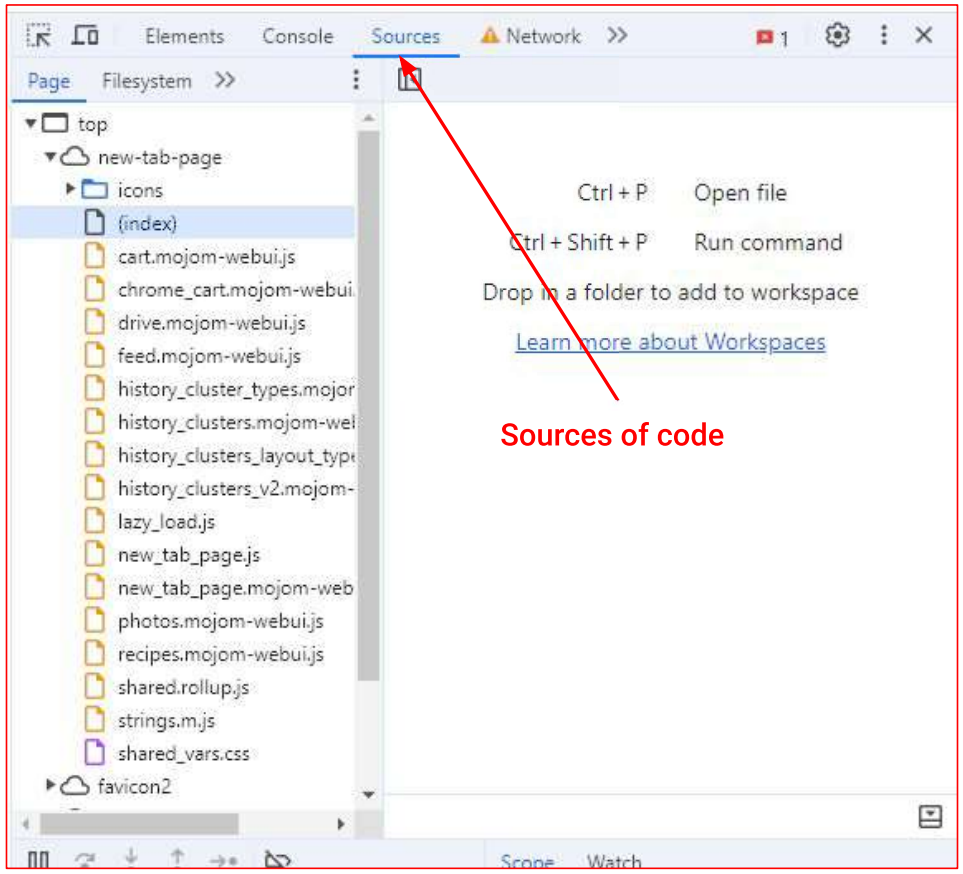
Developer Tools

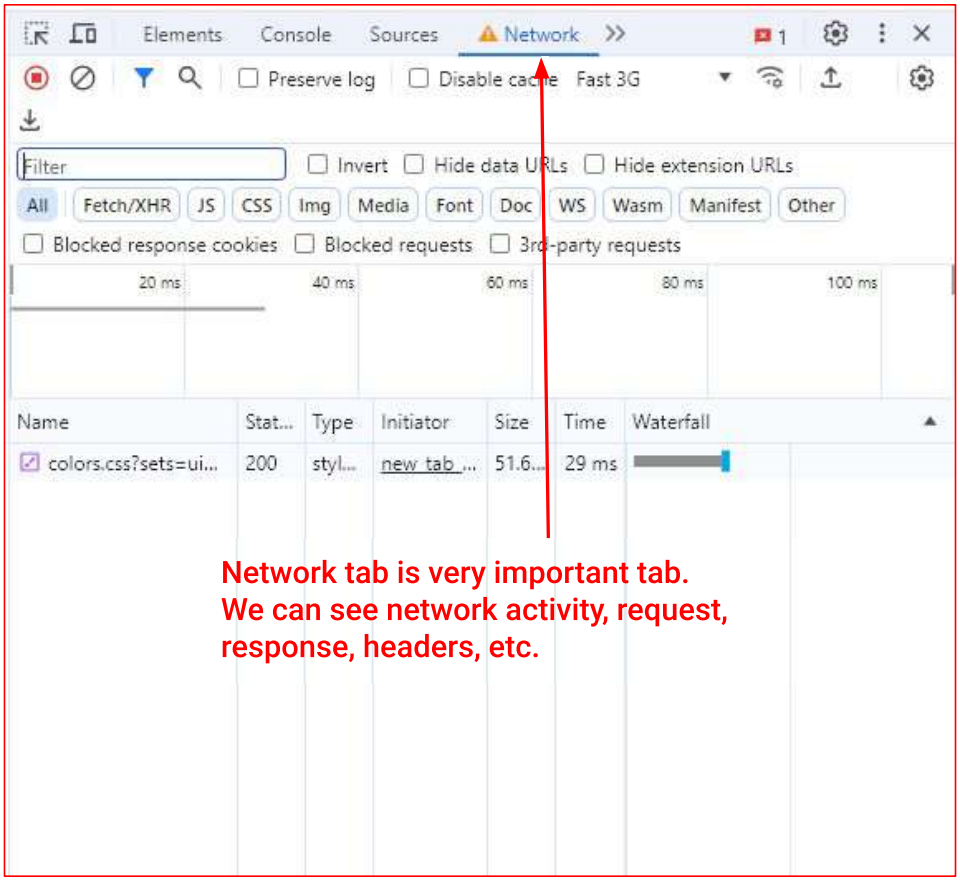
In Chrome browser

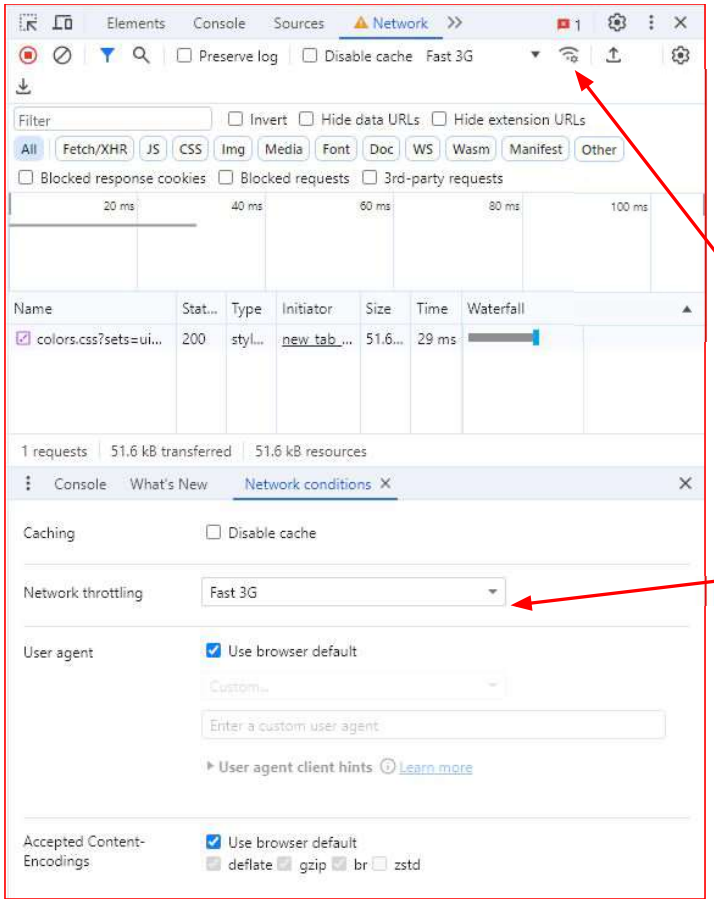





Elements
Console
Sources
Network
top
Default levels
53 Issues: 1 52
Errors are printed in console.
We can write log messages using
console.log() in JavaScript code.

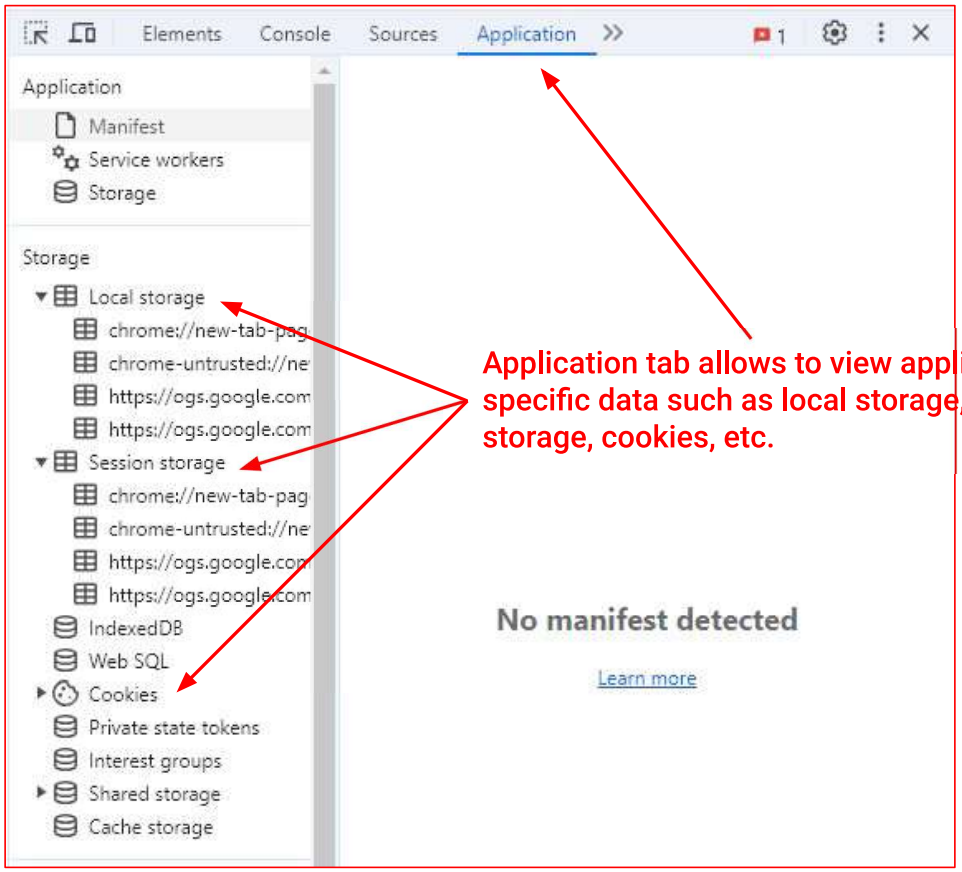

Elements
Console
Sources
Network
Page
Filesystem
top
new-tab-page
icons
(index)
cart.mojom-webui.js
chrome_cart.mojom-webui
drive.mojom-webui.js
feed.mojom-webui.js
history_cluster_types.mojor
history_clusters.mojom-wel
history_clusters_layout_type
history_clusters_v2.mojom-
lazy_load.js
new_tab_page.js
new_tab_page.mojom-web
photos.mojom-webui.js
recipes.mojom-webui.js
shared.rollup.js
strings.m.js
shared_vars.css
favicon2
Ctrl + P Open file
Ctrl + Shift + P Run command
Drop in a folder to add to workspace
Learn more about Workspaces
Sources of code
Scope
Watch

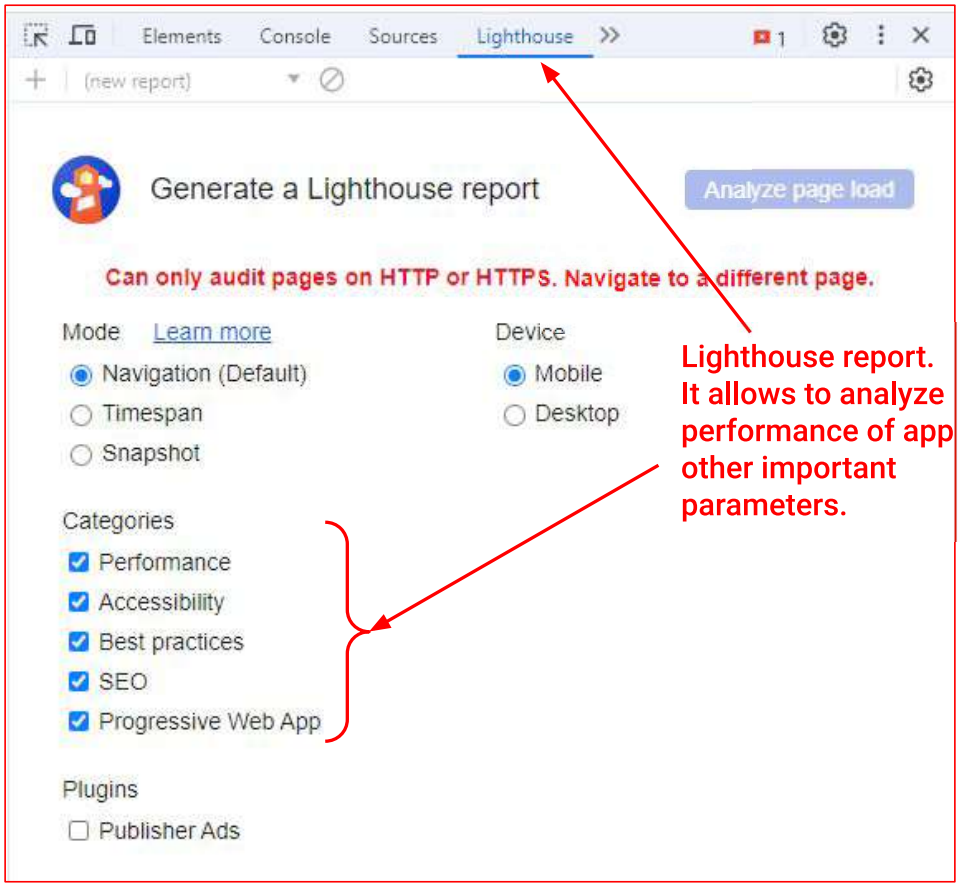




Under network tab, Network throttling allows us to change speed of the network.

Useful to test behaviour of APIs or pages on different networks (2g, 3g, etc.)


Elements
Console
Sources
Application
Application
Manifest
Service workers
Storage
Storage
Local storage
Session storage
IndexedDB
Web SQL
Cookies
Private state tokens
Interest groups
Shared storage
Cache storage
No manifest detected
Learn more
Application tab allows to view application specific data such as local storage, session storage, cookies, etc.


Elements
Console
Sources
Lighthouse
(new report)
Generate a Lighthouse report
Analyze page load
Can only audit pages on HTTP or HTTPS. Navigate to a different page.
Mode
Learn more
Navigation (Default)
Timespan
Snapshot
Device
Mobile
Desktop
Categories
Performance
Accessibility
Best practices
SEO
Progressive Web App
Plugins
Publisher Ads
Lighthouse report. It allows to analyze performance of app and other important parameters.

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.

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