Faculty of Science
Mathematics and Computer Science Department
Lab # 11

Course Code: CMPS 344
Course Name: Software Engineering
Semester: Spring 2023-2024
Instructor: Dr. Bilal Nakhal
TA: Ali Baydoun

Part 1 - Introduction to Git version control:

Introduction

In this part you will become familiar with the basic Git workflow. You will learn to create and manage a source code repository, hosted in GitHub, through the web page interface.

Objectives

The student should:

- Understand the importance of using a version control system in software development.
- Be comfortable with the GitHub flow.

Git using GitHub's web interface

Start by signing up for a free account in GitHub (https://github.com). Once you have logged in, complete the Hello World tutorial (https://guides.github.com/activities/hello-world) that will guide you through your first steps with a distributed version control system.

In this quick start guide, you will:

- Create and use a repository.
- Start and manage a new branch.
- Make changes to a file and push them to GitHub as commits.
- Open and merge a pull request.

Part 2 – Tech Stack:

What is a tech stack?

A stack is like a vertical arrangement of things placed on top of each other. In our case, "things" are technologies used for the development and operation of software.
The tech stack typically includes a programming language, a framework, libraries, databases, and other tools that work together to develop and provide the functionality and features of the application.

Platforms and operating systems are environments in which our software will run. They range from desktop and mobile applications to web platforms. Sometimes we can decide which platforms will be used. However, target platforms are usually predetermined in requirements, especially for client-side software.

Programming languages are formal languages used to write instructions for computers to execute tasks. They provide a way to express algorithms and logic in a human-readable form that computers can understand and execute.

Databases are an integral part of a tech stack, and are structured collections of data that are organized, stored, and managed in a way that allows for efficient information retrieval, updating, and querying.

Frameworks are pre-defined structures that provide a foundation and reusable components for developing software applications. They offer pre-built functionalities, libraries, and conventions that simplify and accelerate development by providing a structure and set of rules to follow. Frameworks often come bundled with libraries specific to the framework's domain, enabling developers to leverage existing code for common tasks.

Libraries are collections of pre-written code modules or functions that provide specific functionalities and can be used by developers to streamline and simplify the development process. Libraries often encapsulate complex operations or offer ready-to-use components, enabling developers to leverage existing code rather than starting from scratch.

Development tools are applications or programs used to assist in software development, testing, deployment, and maintenance. These tools can include integrated development environments (IDEs) like Visual Studio, debugging tools, performance analysis tools, project management tools, and version control systems like Git.

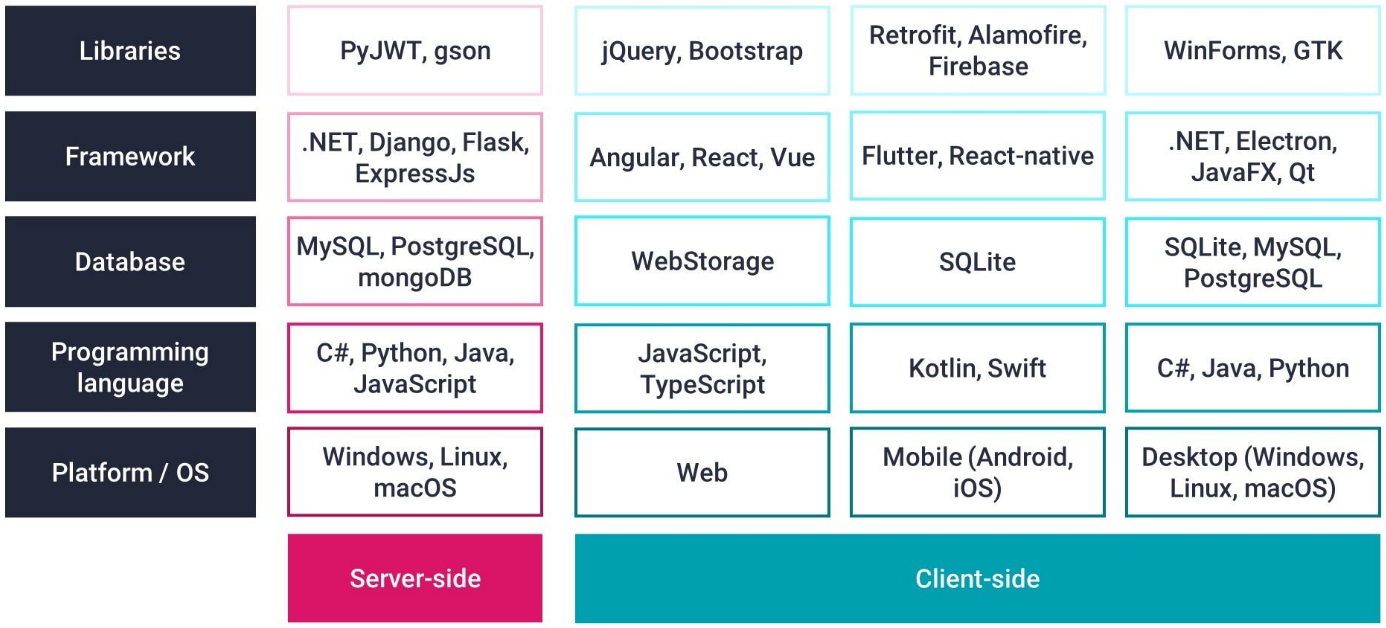
When discussing tech stack, we typically separate apps into **frontend** and **backend** tech stack.

Frontend tech stack refers to the client-facing part of a software application or website, focusing on UI/UX components.
Backend refers to the server-side handling of data storage and processing. For standalone apps, frontend and backend tech stack may not be as distinct.

The below figure provides examples of all the **technologies for each layer of a tech stack**. Due to their differences, the examples are categorized separately for server-side and client-side applications. Additionally, we categorize the examples based on web apps, mobile apps, and desktop platforms for client-side apps. Please note that this list is not exhaustive.

Faculty of Science
Mathematics and Computer Science Department
Lab # 11

Course Code: CMPS 344
Course Name: Software Engineering
Semester: Spring 2023-2024
Instructor: Dr. Bilal Nakhal
TA: Ali Baydoun



Libraries	PyJWT, gson	jQuery, Bootstrap	Retrofit, Alamofire, Firebase	WinForms, GTK
Framework	.NET, Django, Flask, ExpressJs	Angular, React, Vue	Flutter, React-native	.NET, Electron, JavaFX, Qt
Database	MySQL, PostgreSQL, mongoDB	WebStorage	SQLite	SQLite, MySQL, PostgreSQL
Programming language	C#, Python, Java, JavaScript	JavaScript, TypeScript	Kotlin, Swift	C#, Java, Python
Platform / OS	Windows, Linux, macOS	Web	Mobile (Android, iOS)	Desktop (Windows, Linux, macOS)
	Server-side	Client-side		

Here are several project scenarios with varying requirements that can help you choose stack technologies:

1. E-commerce Website:
   Requirements:
   - User registration and authentication
   - Product listing and search functionality
   - Shopping cart and payment integration
   - Order management and tracking

   Stack:
   - Front-end: HTML, CSS, JavaScript, React.js
   - Back-end: Node.js, Express.js
   - Database: MongoDB
   - Payment Integration: Stripe API

2. Social Media Application:
   Requirements:
   - User profiles and authentication
   - News feed and timeline functionality
   - Post creation, sharing, and commenting
   - Real-time notifications and messaging

   Stack:
   - Front-end: HTML, CSS, JavaScript, React.js
   - Back-end: Node.js, Express.js
   - Database: PostgreSQL or MongoDB
   - Real-time functionality: Socket.io or Firebase Realtime Database

3. Data Analytics Platform:
   Requirements:
   - Data ingestion and processing
   - Data visualization and reporting
   - Machine learning model integration
   - Scalable and distributed computing capabilities

   Stack:
   - Front-end: HTML, CSS, JavaScript, React.js or Angular
   - Back-end: Python, Django or Flask
   - Database: PostgreSQL or MySQL
   - Data processing: Apache Spark or TensorFlow

4. Mobile Game Development:
   Requirements:
   - Cross-platform compatibility (Android and iOS)
   - Real-time multiplayer functionality
   - In-app purchases and rewards system
   - Integration with social media platforms

   Stack:
   - Cross-platform framework: React Native or Flutter
   - Game engine: Unity or Cocos2d
   - Backend: Node.js, Express.js
   - Database: Firebase

5. Internet of Things (IoT) Project:
   Requirements:
   - Device connectivity and data collection
   - Cloud-based data storage and processing
   - Real-time monitoring and control
   - Integration with mobile and web applications

   Stack:
   - Device connectivity: MQTT or CoAP
   - Cloud platform: AWS IoT or Microsoft Azure IoT
   - Backend: Node.js, Express.js
   - Database: MongoDB or InfluxDB

6. Content Management System (CMS):
   Requirements:
   - User management and authentication
   - Content creation and publishing
   - Version control and collaboration features
   - Customizable templates and themes

   Stack:
   - Front-end: HTML, CSS, JavaScript, React.js or Angular
   - Back-end: PHP, Laravel or Python, Django

- Database: MySQL or PostgreSQL

7. Online Learning Platform:
   Requirements:
   - User registration and authentication
   - Course management and enrollment
   - Video streaming and conferencing
   - Progress tracking and assessments

   Stack:
   - Front-end: HTML, CSS, JavaScript, React.js or Angular
   - Back-end: Node.js, Express.js or Python, Django
   - Database: PostgreSQL or MongoDB
   - Video conferencing: WebRTC or Zoom API

Part 3: Your Task:

In groups, analyze each scenario and identify potential technology stacks that could be suitable. Encourage discussion and justification for their choices based on the aforementioned factors. Choose the suitable technology stack for your project.

Good Luck