Internet Technologies

Setting up a Local Web Development Environment: Browser / Web Server / Database Server / PHP / PhpMyAdmin / Web Development IDE (editor)

University of Cyprus
Department of Computer Science

Lab Objectives

- Implement web applications from scratch
- By learning and using languages such as HTML, CSS, JavaScript, PHP, Python and Java
- By using databases for storing and retrieving data

Web Applications

- Web application is interactive, server-hosted software, delivered on the World Wide Web (WWW) that can be accessed through a web client
- Web apps consist of frontend (client-side) and backend (server-side) software layers
- Frontend includes everything the user sees and experiences directly
- Backend is invisible to the user; "behind the scenes" part of app

- Frontend communicates with backend using HTTP methods (e.g. GET, POST) to send/receive data

Web Applications Examples

- Website is a collection of publicly accessible web pages containing either documents, images, audio, video, text, or other files that users can access
  - **Static**: Built using front-end languages like HTML, CSS, and JavaScript. They do not involve a database which a user can interact with. They display same content for every user who accesses them.
  - **Dynamic**: In addition to the front-end languages like HTML, CSS, and JavaScript, dynamic websites involve back-end languages such as PHP, JavaScript, Python, ASP to interact with DBs. They are able to display different content for each user
- RESTful API is back-end software (written in PHP, Java, Python) that accepts HTTP msgs (e.g. GET) sent by users that want to access remote resources
  - e.g. OpenWeatherMap REST API allows users retrieve weather data for any location

Entities in Web Application Development

Web Client

chrome

Firefox

Safari

Microsoft Edge

Proprietary program

FRONT-END

HTTP(S)

INTERNET

Web Server / Application Server

APACHE HTTP SERVER

NGINX

Apache Tomcat

node

RAILS

django

Storage System (Database)

mongoDB

MySQL

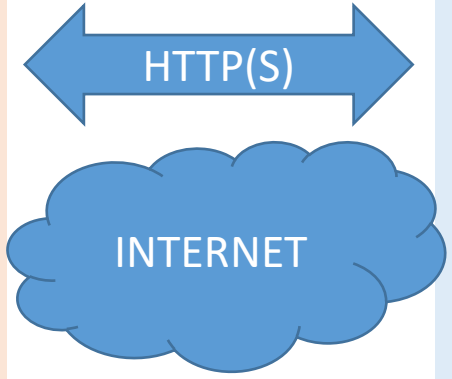
PostgreSQL

BACK-END

Languages in Web Application Development

Web Client

FRONT-END



Web Server / Application Server

Storage System (Database)

BACK-END

Frameworks in Web Application Development

Web Client

Angular by Google

React by Facebook

Vue.js

Apache Cordova

FRONT-END

HTTP(S)

INTERNET

Web Server / Application Server

Laravel

php

CodeIgniter

node.js express

JS

django

Flask web development, one drop at a time

spring

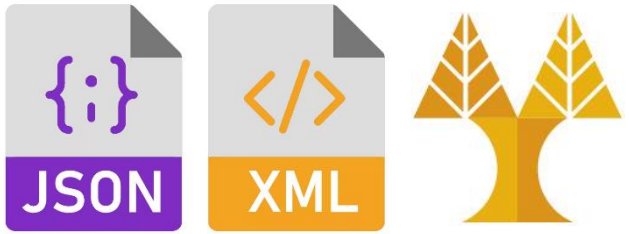
BACK-END

Storage System (Database)

Servers: Web vs Database vs Application

- Web server: Handles HTTP requests (e.g. GET, POST) for static & dynamic websites sent by users via browser
  - Example web servers: Apache, Nginx, Microsoft IIS
- Database server: Stores data and handles database queries (e.g. SELECT)
  - Example database servers: MySQL, MariaDB, PostgreSQL, MongoDB
- Application server: Sits mainly in front of a database server and basically exposes data from databases. Application server hosts applications (e.g. written in Java, PHP, .NET) that are accessed using APIs. Exposed data is returned in JSON, XML and HTML format.
  - Example application servers: Apache Tomcat, Jetty, Websphere, Jboss, Glassfish (Java), Zend Server (PHP), Mono (.NET)

XML vs JSON



- Both used to format data exchanged between client and server
- XML was developed in 1997 and uses identifying tags similar to HTML
- JSON was developed in 2001, derived from JavaScript and can be condensed with less characters to be very lightweight

XML

```
<employees>
  <employee>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName>
    <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName>
    <lastName>Jones</lastName>
  </employee>
</employees>
```

JSON

```
{"employees":[
  { "firstName":"John", "lastName":"Doe" },
  { "firstName":"Anna", "lastName":"Smith" },
  { "firstName":"Peter", "lastName":"Jones" }
]}
```

Apache HTTP Server

- Free, open source HTTP Server.
- Developed and maintained by Apache Group – first released in 1995.
- Its name comes from the Indian tribe famous for their strength and endurance in battle.
- Provides load balancing and caching.
- Runs on UNIX and Windows platforms. UNIX version is the most stable and most widely used.
- Installing and maintaining is easy; low complexity & low difficulty of administration.
- https://httpd.apache.org/

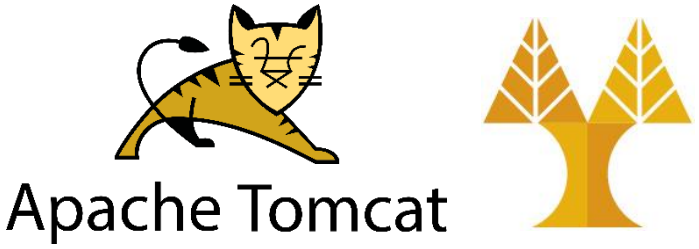
NGINX HTTP Server

- Free, open-source, high-performance Web Server and reverse proxy, as well as an IMAP/POP3 proxy server.
  - Commercial support and training available
- Provides load balancing and HTTP caching.
- Unlike traditional servers, NGINX doesn't rely on threads to handle requests. Instead it uses a much more scalable event-driven (asynchronous) architecture.
- Predictable performance (RAM usage, CPU usage, latency) under high loads.
- https://www.nginx.com/

Node.js

- Open-source and cross-platform JavaScript runtime environment for developing custom server-side network applications in JavaScript:
  - Web Server just like Apache or Nginx, TCP server, DNS server, Web chat app, online games, collaboration tools or anything which sends updates to the user in real time
- Node.js runs the V8 JavaScript engine (written in C++), the core of Google Chrome, and executes JavaScript code outside of the browser
- NPM (Node Package Manager) comes bundled with Node.js installation and it is the default package manager for Node.js
  - Manages dependencies for an application
  - Allows users to install Node.js applications that are available on the npm registry
- https://nodejs.org/en/download/

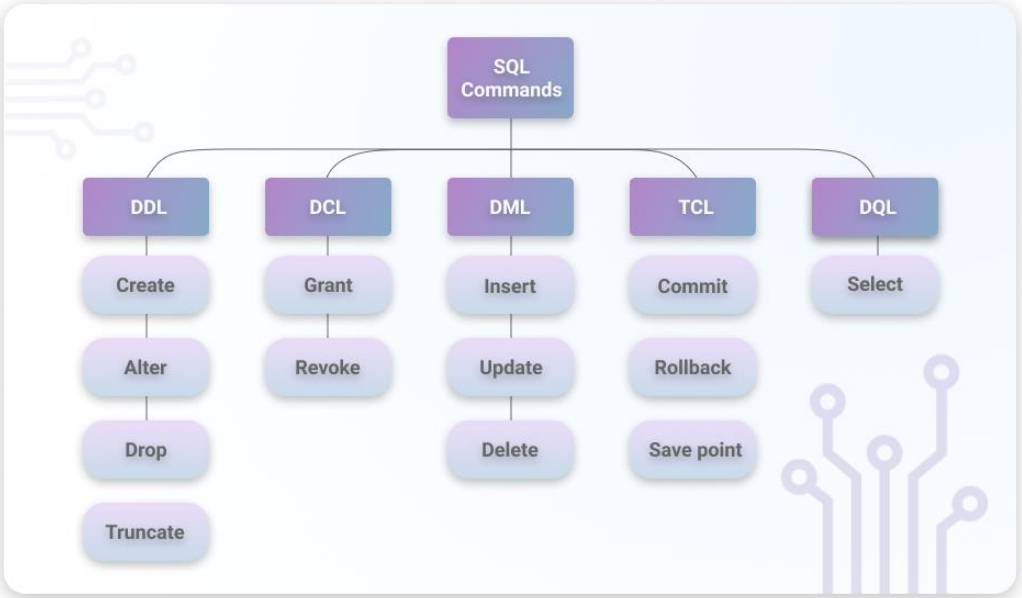
Apache Tomcat



- Free, open-source, Web Application container
- Written in Java (OS independent)
- Tomcat is used to run Servlet and Java Server Pages (JSP) web applications
  - Servlet is a server-side Java program (.java) that handles client requests (HTTP GET, POST, etc.).
  - JSP allows for inserting Java code into HTML pages. JSP files have the .jsp extension.
- https://tomcat.apache.org/

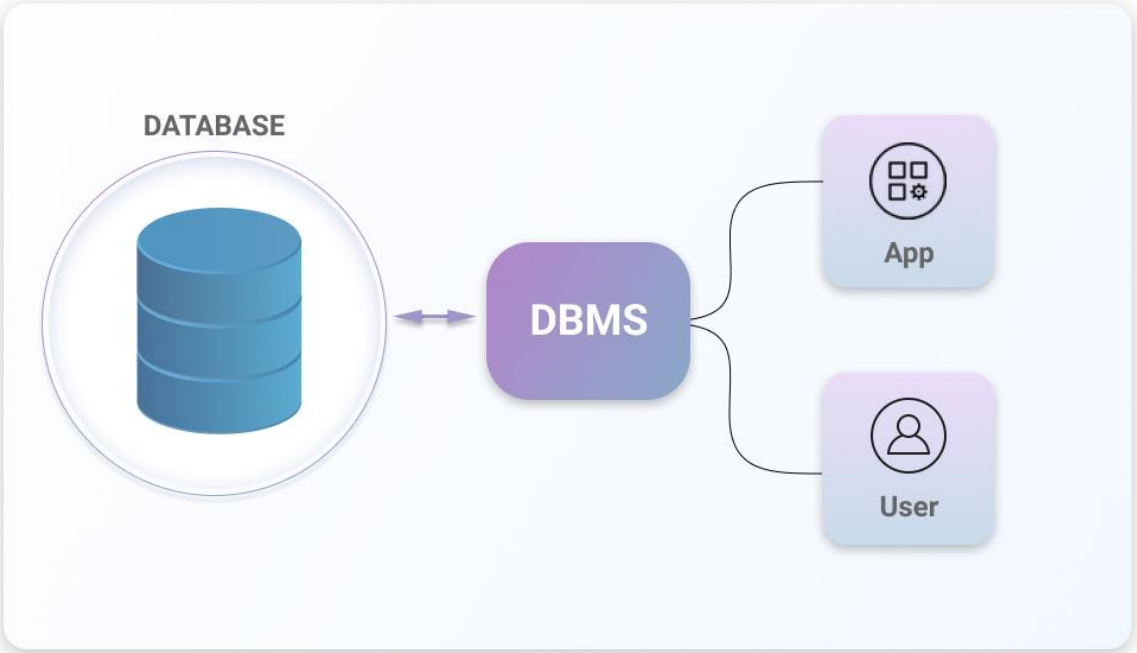
Databases: Relational vs NoSQL

- Relational databases store data in a structured tabular (schema-based) format: using rows and columns
- Why ‘relational’ though? Because tables in a relational database can be linked or joined or related based on common data (public/private keys).
- SQL is a computer language used by most relational database management systems (RDBMS) to store, manipulate, and retrieve data stored in the tabular format.



Databases: Relational vs NoSQL

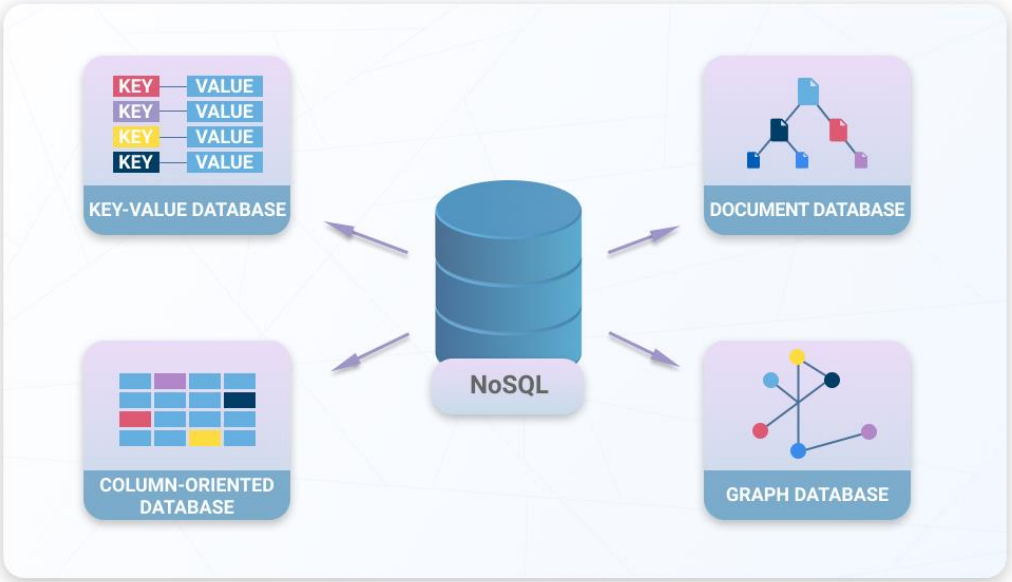
- RDBMS stands for Relational Database Management System and represents software that is used to manage, manipulate, query, and retrieve data stored in a relational database.



Popular RDBMS

Databases: Relational vs NoSQL

- Aka "**N**ot **o**nly **SQL**"
- Non-tabular (schema-less) databases
- Without concept of relationships (joins)
- Store data differently than relational DBs
  - Store data in documents similar to JSON
  - Store data in key-value pairs
  - Store data in tables, rows, and dynamic columns
    - the names and the format of the columns can vary from row to row within the same table
  - Store data in nodes and edges as a graph
    - Nodes typically store information about people, places, and things, while edges store information about the relationships between the nodes.

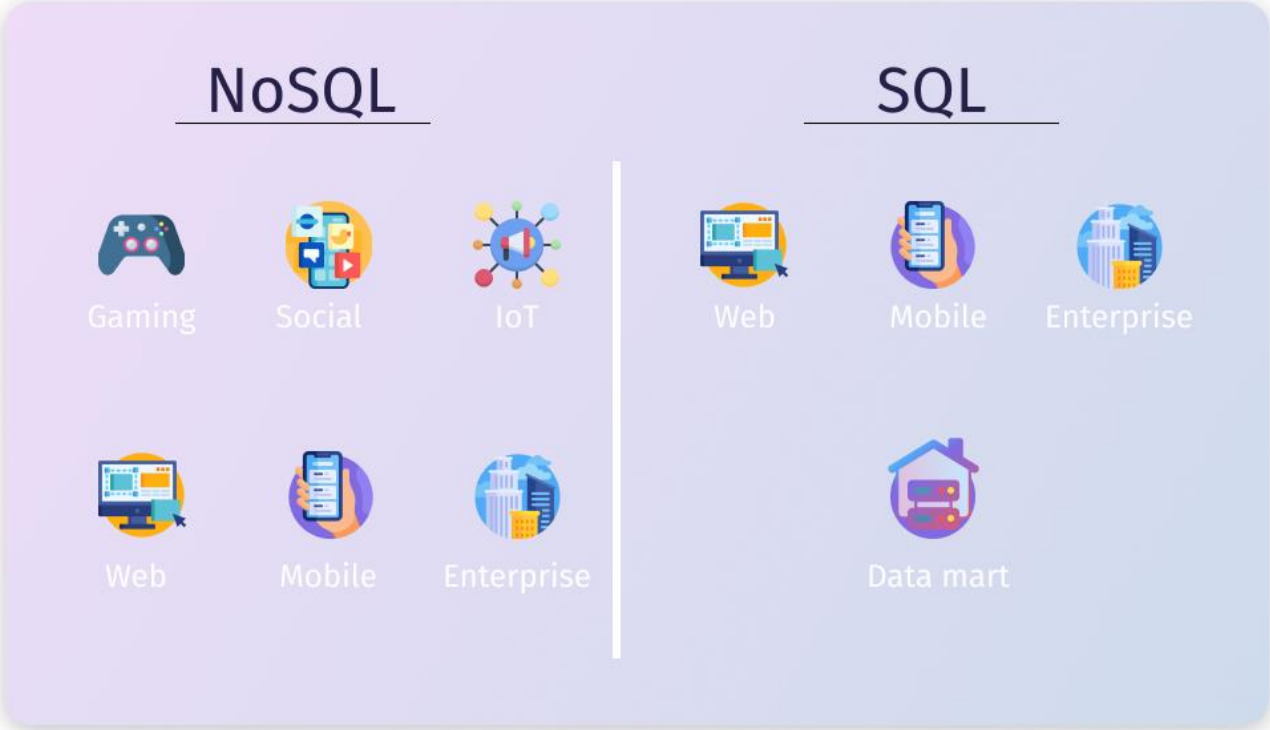


Databases: Relational vs NoSQL

- **Document databases**: MongoDB, CouchDB.
- **Key-value stores**: Cassandra, Azure, Redis, DynamoDB, LevelDB, Riak.
- **Wide-column stores**: HBase, BigTable, HyperTable.
- **Graph databases**: Amazon Neptune, Polyglot, Neo4J.

Databases: Relational vs NoSQL – When to use each?

- Depends on the type of data you're going to store and how you're going to store it
- If your data is generally structured, a relational database is likely the best choice.
- Non-relational databases are primarily used to store and process Big Data for real-time web apps.



MySQL

- Open-source, enterprise-level, multi-threaded database management system
  - commercial license available (MySQL is now owned by Oracle)
- MySQL Document Store allows developers to work with both SQL relational tables and schema-less (No-SQL) JSON collections (as of MySQL 8.0).
- Works on many different platforms (UNIX, Windows, SaaS on Cloud).
- Handles large databases.
- https://www.mysql.com/

MariaDB

- Open-source, community-developed, free relational database management system
- Development is led by some of the original developers of MySQL, who forked it due to concerns over its acquisition by Oracle Corporation in 2009
- Intended to maintain compatibility with MySQL – however, new features are diverging
- https://mariadb.com/ (enterprise server) and https://mariadb.org/ (community server)

PostgreSQL

- Open-source, enterprise-level, multi-process, relational database management system
- Supports SQL querying
- Supports native JSON data type (within a relational table) since version 9.2
  - Provides many functions and operators for manipulating JSON data.
- Works on many different platforms (UNIX, Windows, SaaS on Cloud).
- Most geospatial features of any open-source database
  - Geographic objects, geographic queries
- Designed for high volume environments
- https://www.postgresql.org/

MongoDB

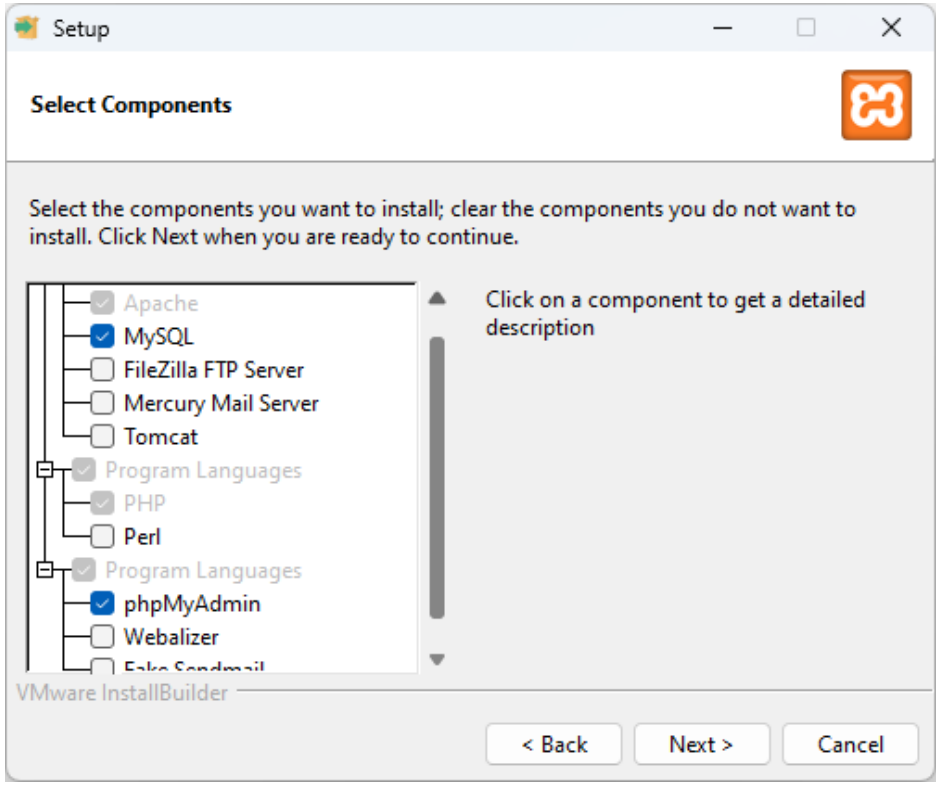
- Open-source, scalable, document-oriented (No-SQL) database
- Documents stored in collections (a collection is similar to a table in relational databased)
  - Created on the fly when referenced for the first time
- Rich document-based queries
- No joins and no multi-document transactions
- Provides geospatial features
- https://www.mongodb.com

Web Application Stack Frameworks

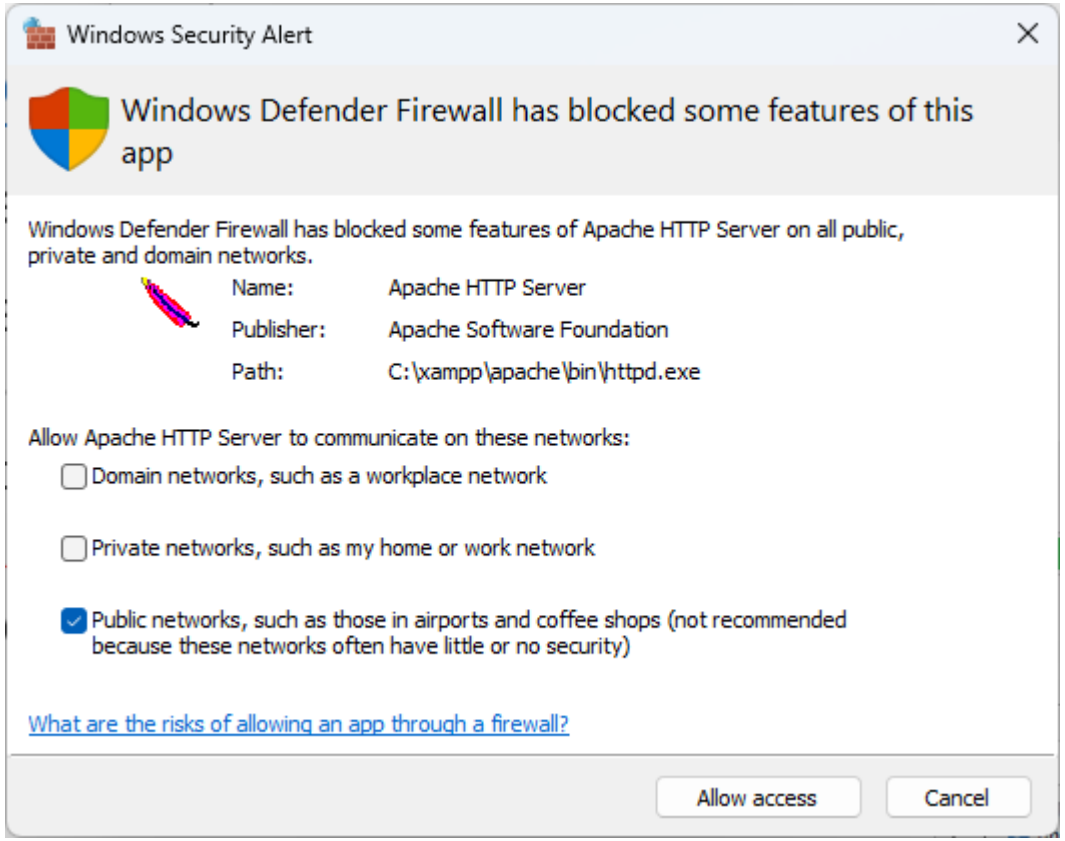
-  XAMPP
  - **X** ("cross-platform"), **A**pache, **M**ariaDB, **P**HP, **P**erl
    - MySQL was replaced with MariaDB in 2015
  - Download: XAMPP home
- LAMP
  - **L**inux, **A**pache, **M**ySQL, **P**HP / Perl / Python
- WAMP
  - **W**indows, **A**pache, **M**ySQL, **P**HP / Perl / Python
  - Download : WampServer, easyPHP
- MAMP
  - **M**acOS, **A**pache, **M**ySQL, **P**HP / Perl / Python

Install XAMPP Stack on your machine

- Go to https://www.apachefriends.org/
- Download the appropriate installer based on your OS
- Install Apache, MySQL, PHP and phpMyAdmin only
  - No need for Filezilla FTP Server, Mercury Mail Server, Apache Tomcat, Perl language, Webalizer, Fake Sendmail
  - For Windows install on C:\xampp

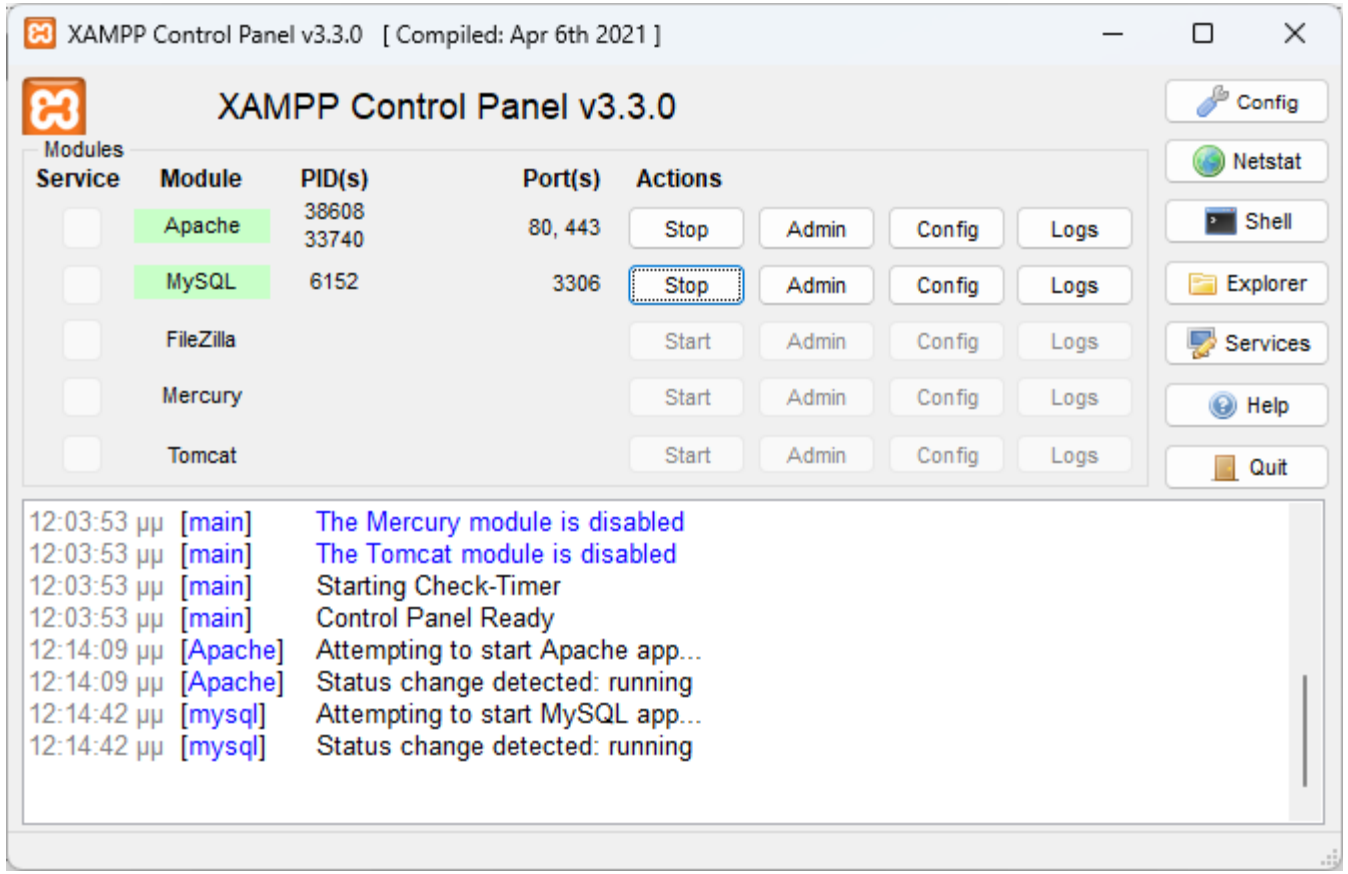


Allow access to Apace HTTP Server



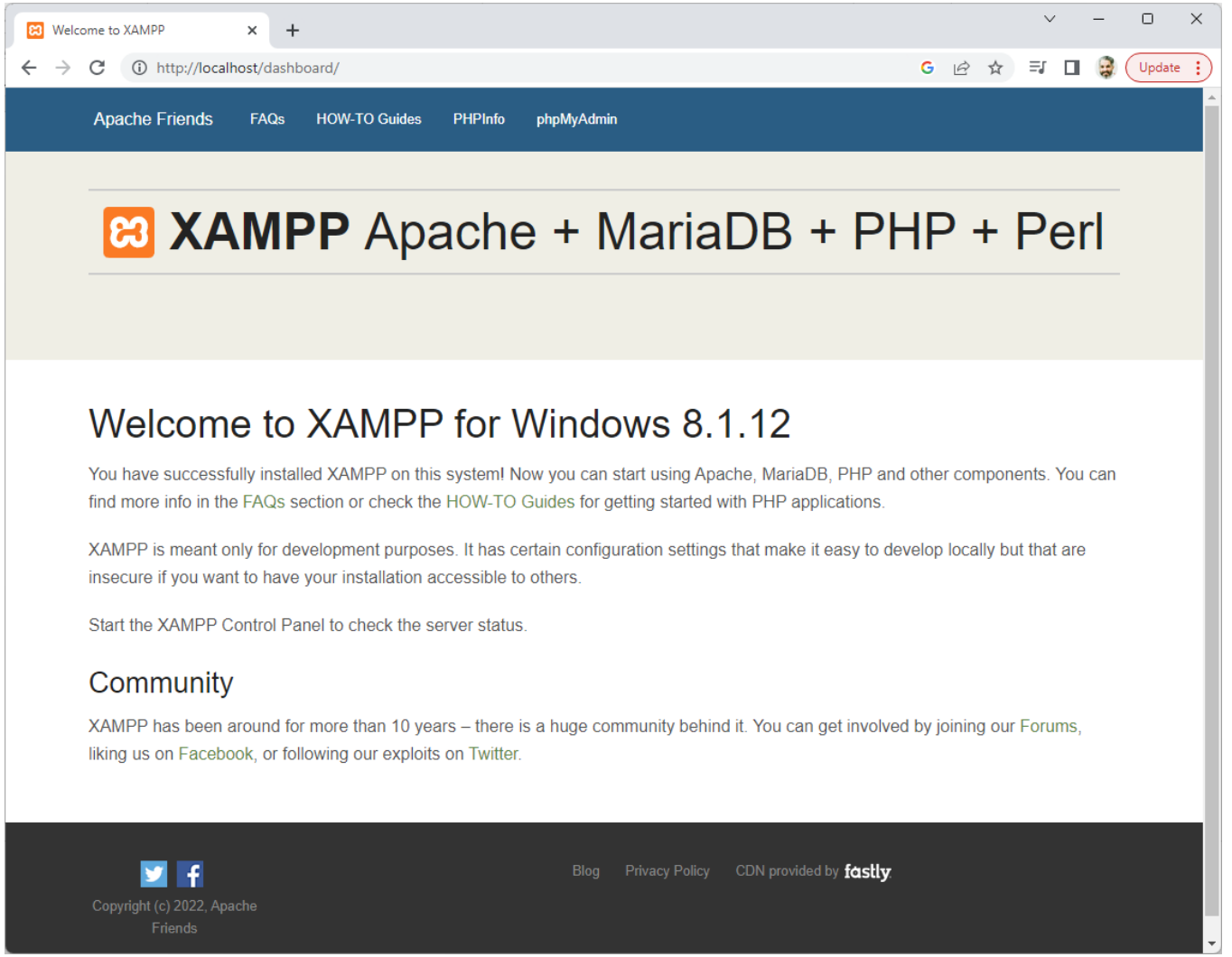
XAMPP Control Panel

- Start Apache
- Start MySQL



Check if local Web Server is running…

- Open browser
- Navigate to http://127.0.0.1 or http://localhost
- We can use https instead of http but we need to setup a trusted certificate to remove the “Not secure” message



Create a simple web page in local Web Server

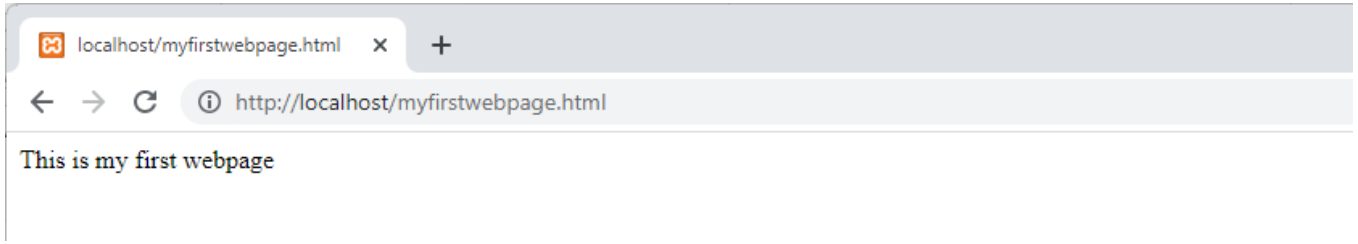
- Go to `C:\xampp\htdocs`
- Create an html file called `myfirstwebpage.html` with the following content:

```
<p>This is my first webpage</p>
```

- Files can be created using `notepad` command or a dedicated IDE (see next slides)
- Navigate to this webpage using your browser

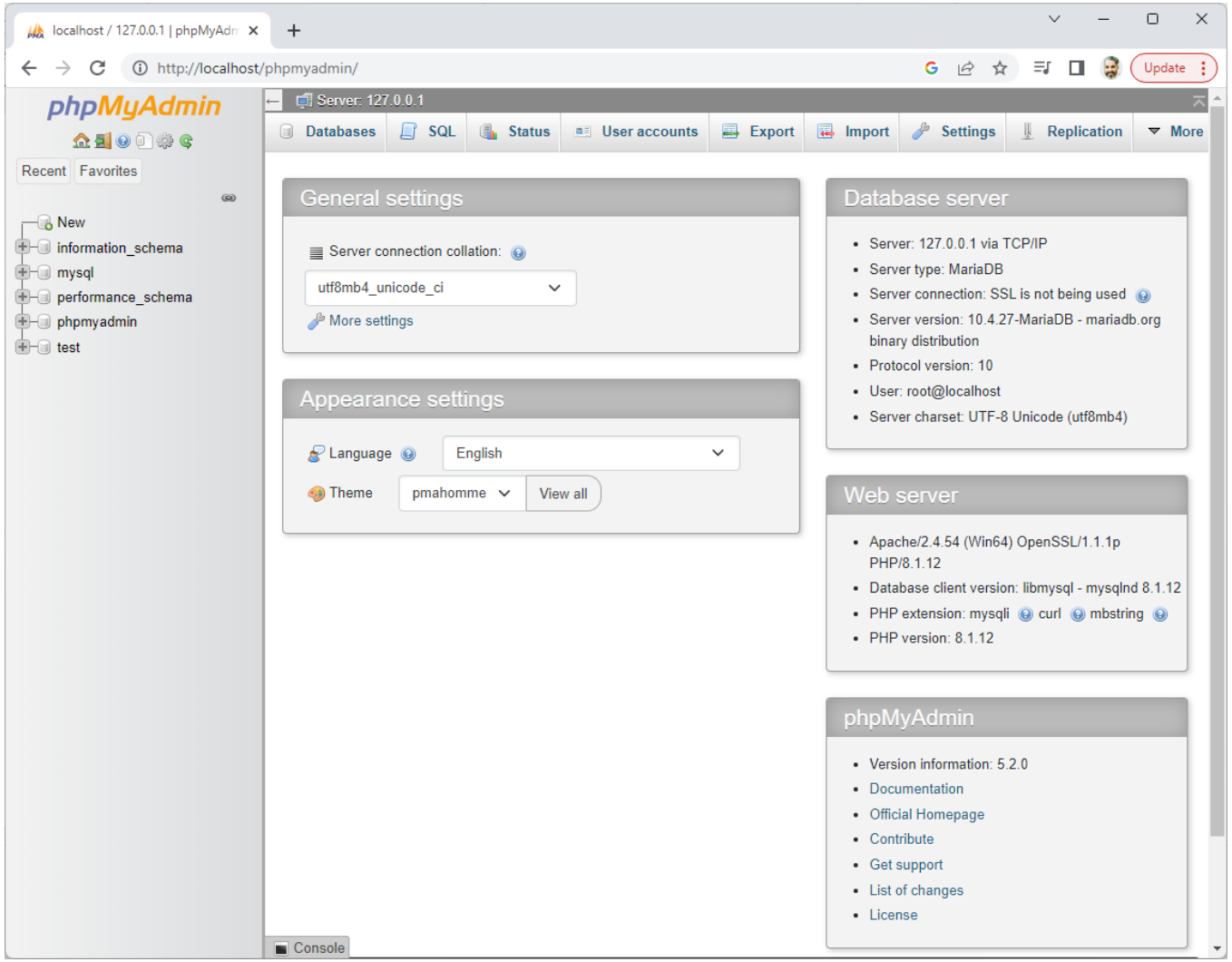
Create a simple web page in local Web Server

- Open browser
- Navigate to http://localhost/myfirstwebpage.html



Check if local MySQL is running…

- Use phpMyAdmin tool
- Open browser
- Navigate to http://127.0.0.1/phpmyadmin or http://localhost/phpmyadmin



phpMyAdmin

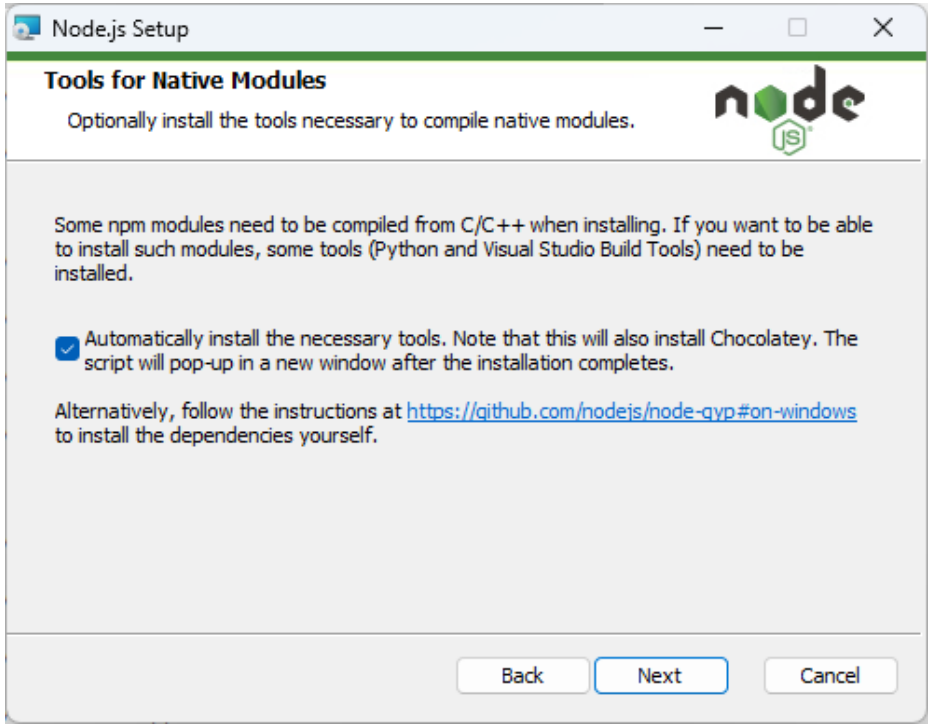
- Free software tool written in PHP, intended to handle the administration of MySQL or MariaDB via a browser
- Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the graphical user interface, while you still have the ability to directly execute any SQL statement

Gain access to remote Web servers

- Normally, a real-life web application is running on a remote server
- **Command-line (terminal) access** on a remote server is obtained via SSH (secure shell)
  - Usage: create folders, change permissions on files/folders, configure web server
  - Available tools: putty, X2Go (includes GUI), Remote Desktop Connection (Windows)
- **Access to upload/download** data on a remote server is obtained via (secure) FTP (File Transfer Protocol)
  - Usage: create/delete/update folders, upload/download folders & files
  - Available tools: FileZilla, WinSCP
- Recommended (Windows OS) tool for **both tasks**: MobaXterm (free)

Install Node.js on your machine (optional)

- Go to https://nodejs.dev/en/download/
- Download the appropriate installer based on your OS
- Double click the executable
  - Opt-in for installing the tools for native (C++) modules that come along with Node.js
  - To check if Node is installed, open the Windows Command Prompt (cmd), Powershell or a similar command line tool, and type `node -v` This should print a version number, so you'll see something like this v18.13.0

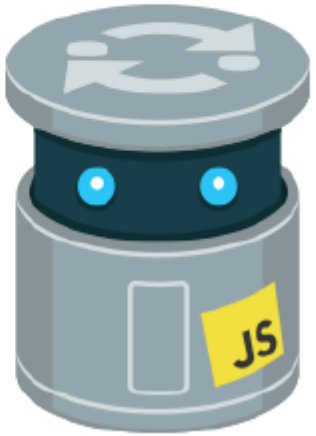


Web Development IDEs

- Visual Studio Code  HTML, CSS, JavaScript, PHP) ← recommended
- Sublime Text 3  (HTML, CSS, JavaScript, PHP)
- PHPStorm  (HTML, CSS, JavaScript, PHP, and more)
- WebStorm  (HTML, CSS, JavaScript, React, and more)
- NetBeans  (HTML, JavaScript, PHP)

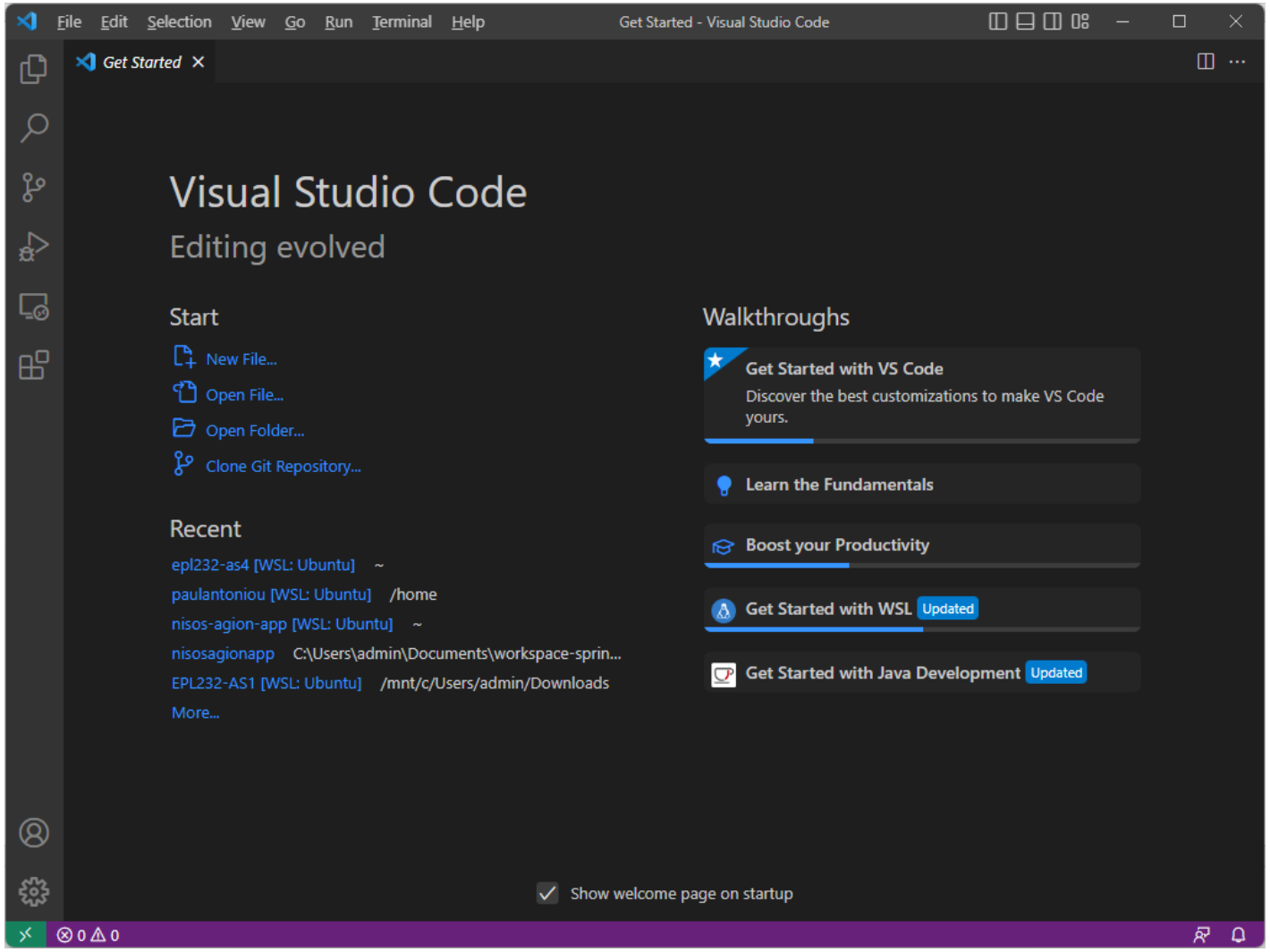
Online Development Tools

- Provide ease of access and development ready environment for front end web development. Most widely used are:
  - Codepen.io
  - CSSDesk
  - JS Bin
  - JSFiddle
  - Dabblet
  - D3 Playground



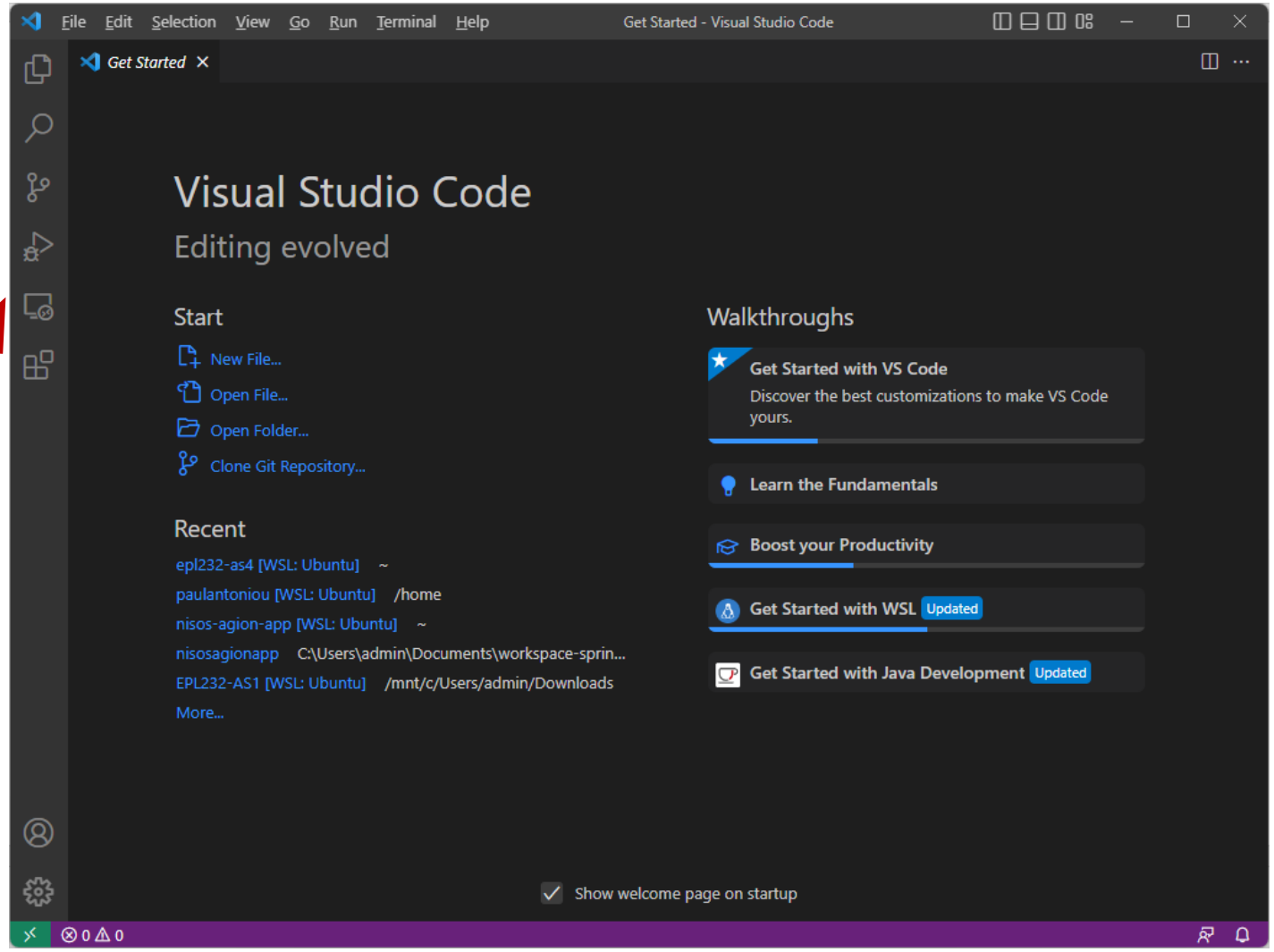
Install VS Code (recommended)

- If you don't yet have VS Code installed on your machine, head to code.visualstudio.com to download it
- Double click (on Windows) executable installer file to install VS Code
- Launch VS Code



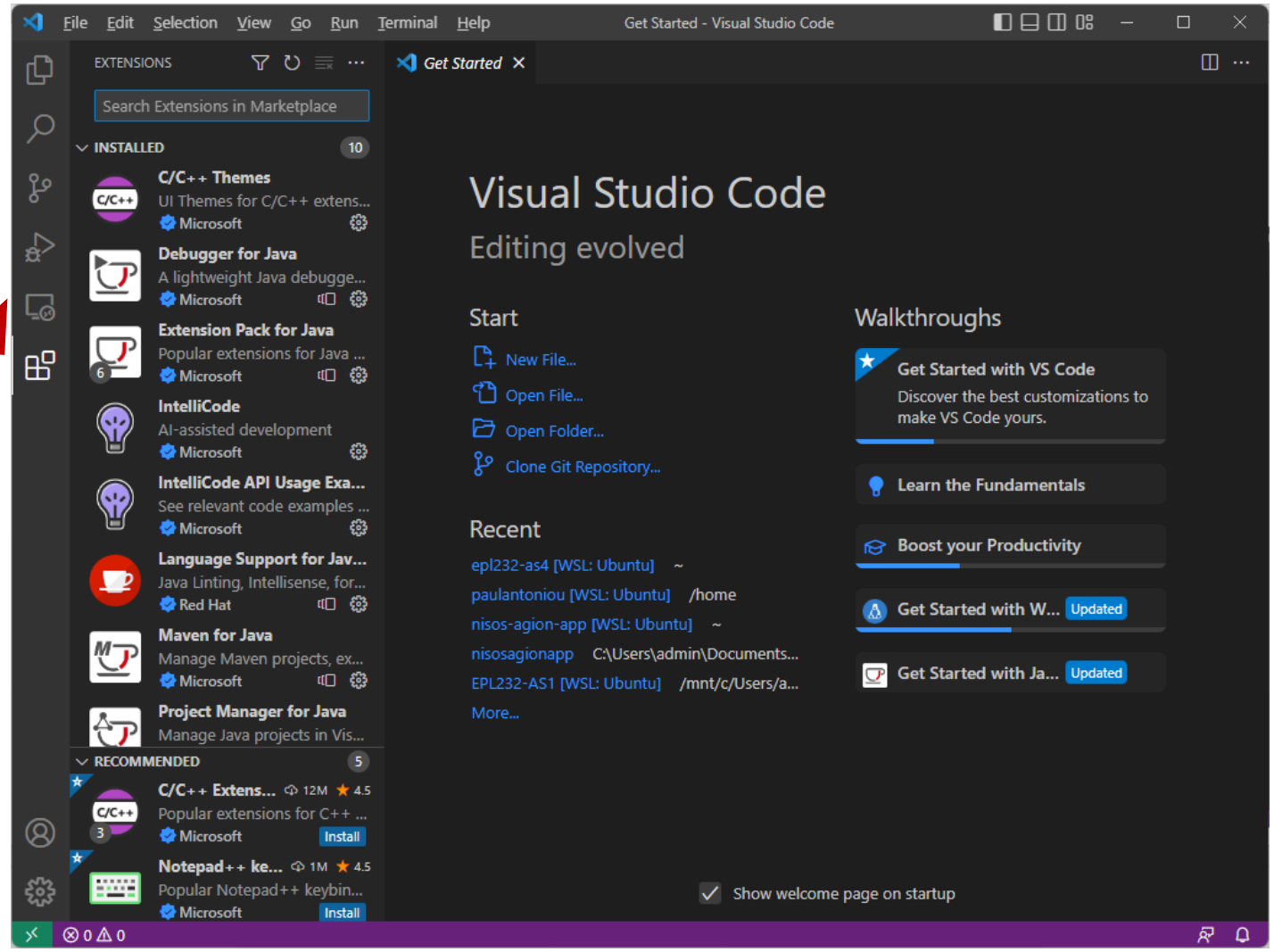
Install Web development extensions

- Open **Extensions** view



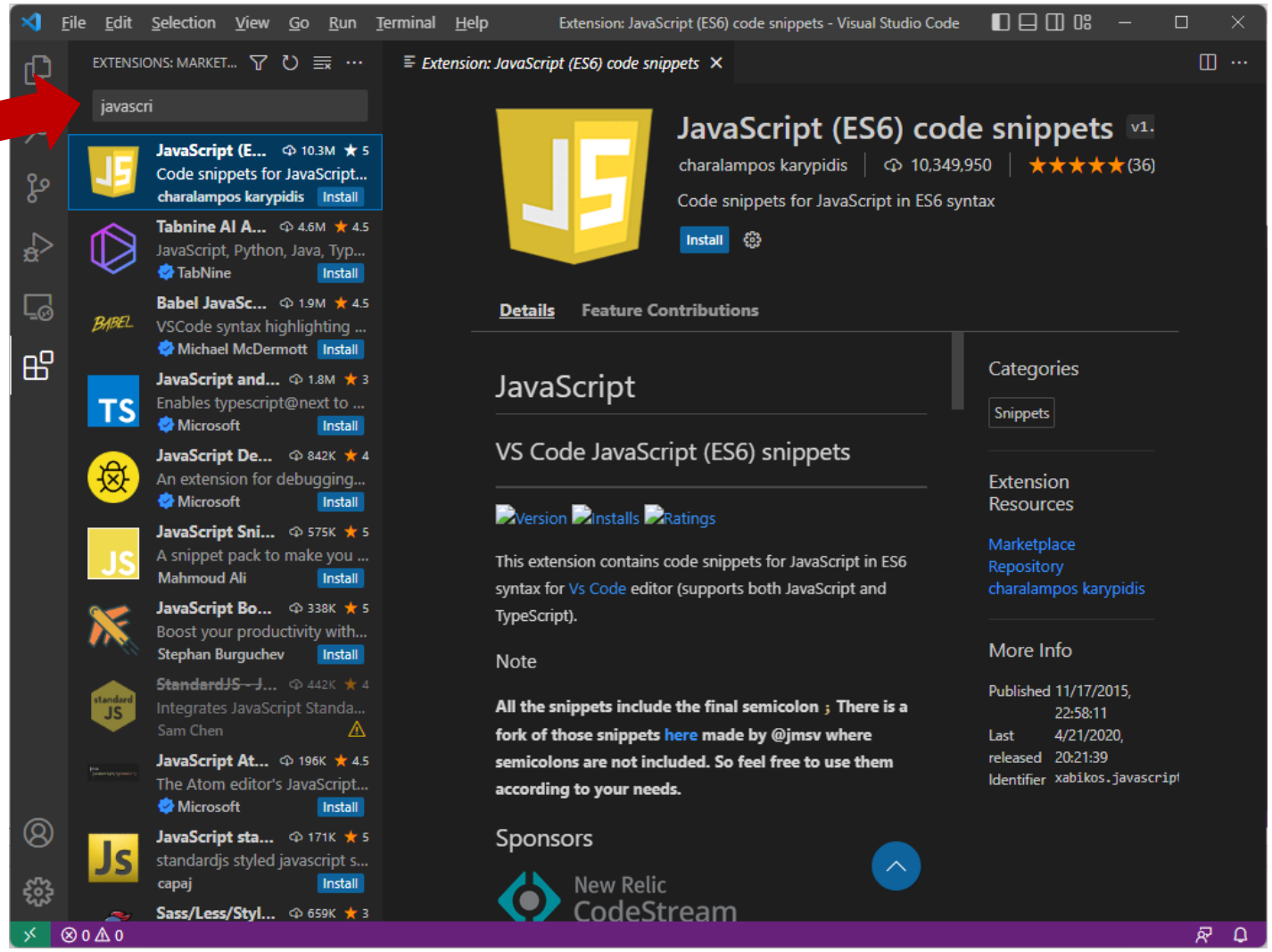
Install Web development extensions

- Open **Extensions** view



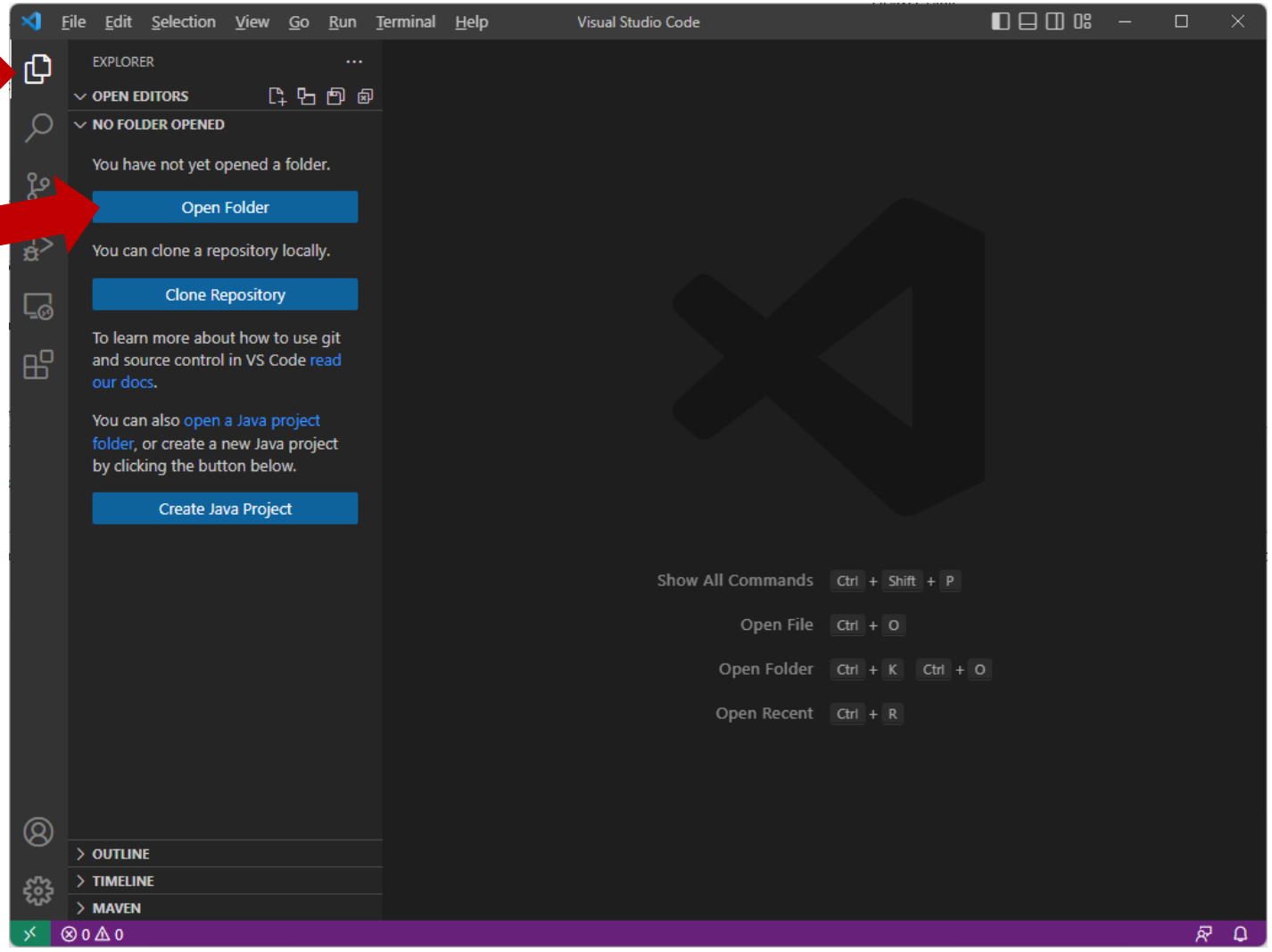
Install Web development extensions

- Type and install:
  - JavaScript (ES6) Code Snippets
  - PHP Intelephense
  - ESLint
  - CSS Peek
  - Auto Close Tag
  - Prettier – Code formatter
  - Path intellisense
  - Remote Development
  - Live Preview



Importing or Creating a New Web Project

- Click on the first (upper) icon on the left menu bar to access the file explorer
- Click **Open Folder** button and select your project's directory, e.g. C:\xampp\htdocs\dashboard
- In case you are creating a new project, first you need to create a new folder e.g. C:\xampp\htdocs\mywebsite and then select (Open) that as your project directory.



Loading a Web Project

Live Preview,
see next slide

- The explorer window will now show the project's directory tree, giving you quick access to files and directories on your project.
- You can view and edit the source code of a file by clicking on it

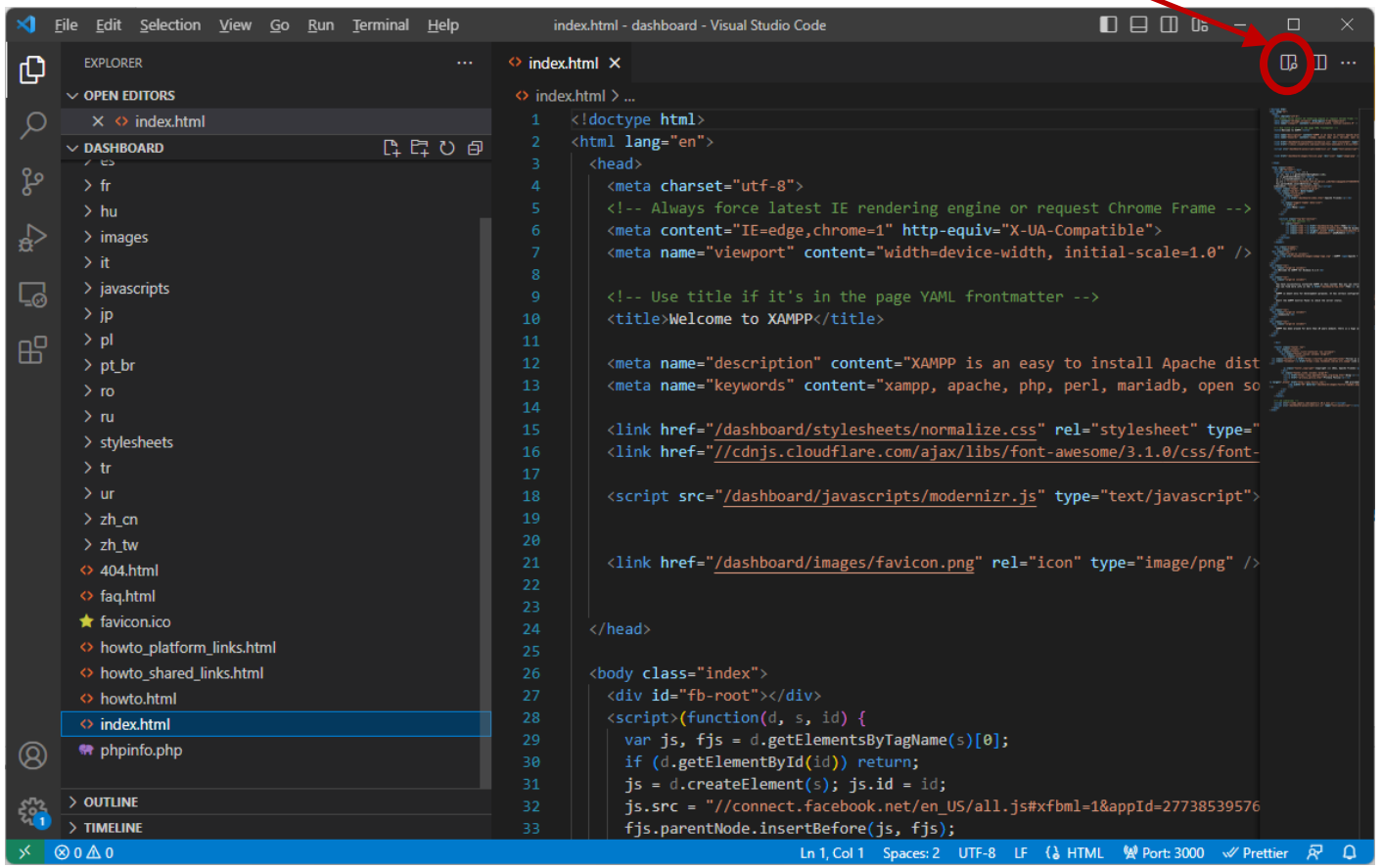
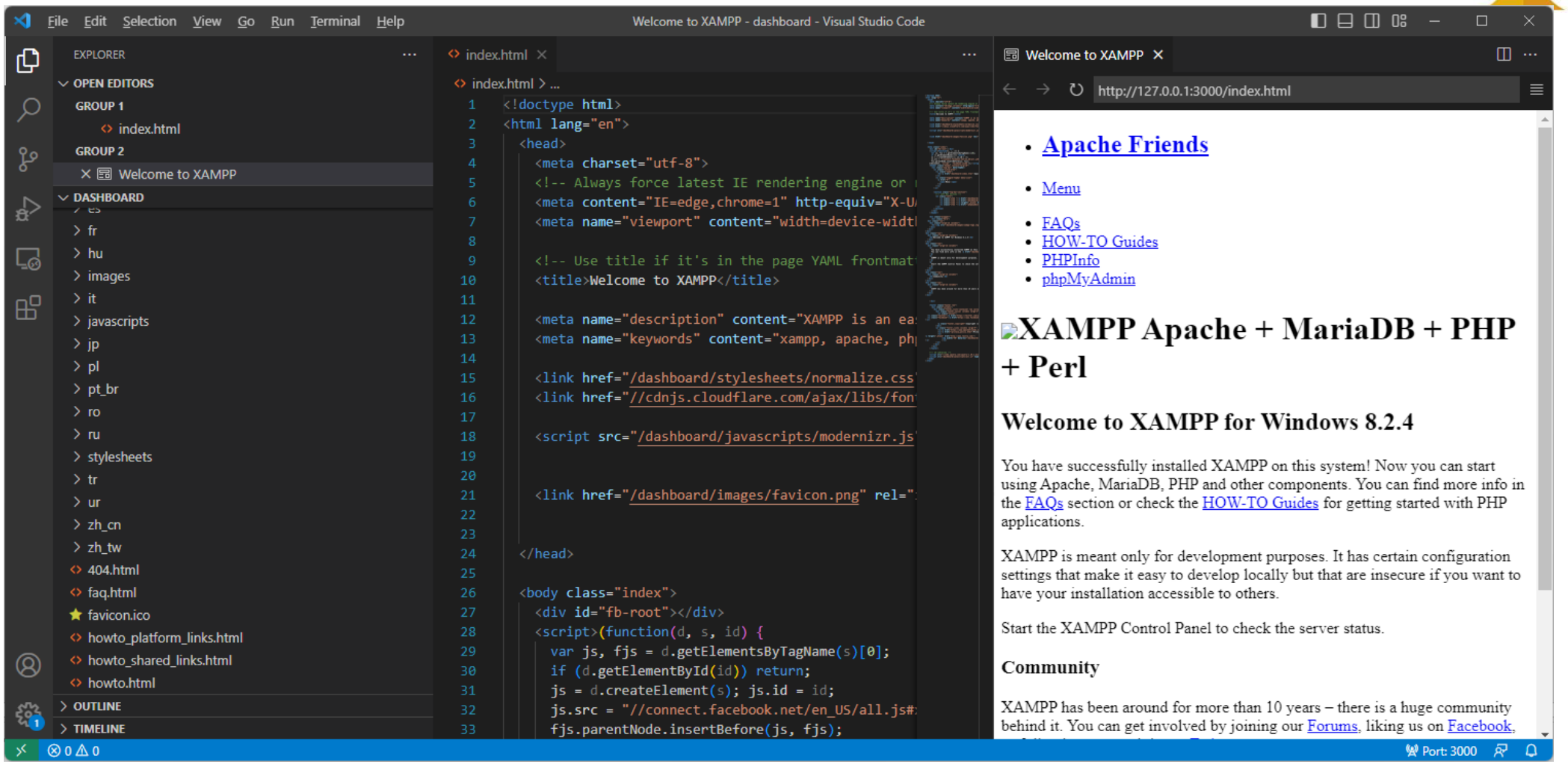


Figure: VSCode → Open folder → C:\xampp\htdocs\dashboard
Double click on index.html

Preview Web Project using Live Preview



If you have the “Live Preview” extension installed, you can preview your HTML files (VSCode hosts a local server)
Appropriate for front-end development (HTML, CSS, JavaScript)