

**SRINIVAS UNIVERSITY MANGALORE**  
**INSTITUTE OF COMPUTER SCIENCE AND INFORMATION SCIENCES**  
**Course Name: MCA II Year: Semester Name: III Semester**  
**Subject Name: E - COMMERCE DEVELOPMENT USING MAGENTO**

**Unit- I**

**What is E-commerce? Types of E-commerce in real world, Different E-Commerce Platform, Top websites developed using Magento, what is Magento? Why to use Magento? Version, Features, Advantages and Disadvantages of Magento, Installation steps for Magento, System Requirements for in installation, Need Of composer, Magento Architecture and Framework, Block diagram, Magento Folder Structure, Magento Design Concepts and Terminology with scenarios, Global-Website-Store methodology, Planning for multiple stores, Magento2 Components**

## Introduction to E – Commerce

Electronic commerce (e-commerce) refers to companies and individuals that buy and sell goods and services over the internet. E-commerce operates in different types of market segments and can be conducted over computers, tablets, smartphones, and other smart devices. Nearly every imaginable product and service is available through e-commerce transactions, including books, music, plane tickets, and financial services such as stock investing and online banking.

### KEY TAKEAWAYS

- E-commerce is the buying and selling of goods and services over the internet.
- It is conducted over computers, tablets, smartphones, and other smart devices.
- Almost anything can be purchased through e-commerce today, which makes e-commerce highly competitive.
- It can be a substitute for brick-and-mortar stores, though some businesses choose to maintain both.
- E-commerce operates in several market segments including business-to-business, business-to-consumer, consumer-to-consumer, and consumer-to-business.

### What is E- Commerce?

E-commerce (electronic commerce) is the buying and selling of goods and services, or the transmitting of funds or data, over an electronic network, primarily the internet. These e-commerce transactions typically fall within four types: business-to-business (B2B), business-to-consumer (B2C), consumer-to-consumer or consumer-to-business.

The terms *e-commerce* and *e-business* are often used interchangeably. The term *e-tail* is also sometimes used in reference to the transactional processes that make up online retail shopping.

### How does E-commerce work?

E-commerce is powered by the internet. Customers use their own devices to access online stores. They can browse products and services those stores offer and place orders.

As an order is placed, the customer's web browser communicates back and forth with the server hosting the e-commerce website. Data pertaining to the order is relayed to a central computer known as the order manager. The data is then forwarded to databases that manage inventory levels; a merchant system that manages payment information using payment

processing applications, such as PayPal; and a bank computer. Finally, it circles back to the order manager. This ensures store inventory and customer funds are sufficient for the order to be processed.

After the order is validated, the order manager notifies the store's web server. It displays a message notifying the customer that their order has been processed. The order manager then sends order data to the warehouse or fulfillment department, letting it know the product or service can be dispatched to the customer. At this point, tangible and digital products are sent to the customer, or access to a service is granted.

Platforms that host e-commerce transactions include online marketplaces that sellers sign up for, such as Amazon; software as a service (SaaS) tools that let customers rent online store infrastructures; and open source tools that companies manage using their in-house developers.

## **Types of E- Commerce in real world**

E-commerce, or electronic commerce, refers to the buying and selling of goods and services over the internet. There are several types of e-commerce models that exist in the real world, each catering to different needs and preferences of consumers and businesses. Here are some common types of e-commerce along with examples:

### **1. Business-to-Consumer (B2C):**

- Business-to-consumer e-commerce involves transactions between a business and individual consumers.
- Example: Amazon.com is a prime example of B2C e-commerce. Consumers can purchase a wide range of products directly from Amazon's website.

### **2. Business-to-Business (B2B):**

- Business-to-business e-commerce involves transactions between businesses.
- Example: Alibaba.com facilitates B2B transactions by connecting businesses with manufacturers, wholesalers, and suppliers.

### **3. Consumer-to-Consumer (C2C):**

- Consumer-to-consumer e-commerce involves transactions between individual consumers.
- Example: eBay is a popular C2C e-commerce platform where individuals can buy and sell a variety of products to each other.

### **4. Consumer-to-Business (C2B):**

- Consumer-to-business e-commerce involves transactions where individuals offer products or services to businesses.

- Example: Freelance platforms like Upwork or Fiverr allow individuals to offer their skills and services to businesses in need of specific tasks or projects.

#### **5. Peer-to-Peer (P2P):**

- Peer-to-peer e-commerce involves direct transactions between individuals without the involvement of a centralized business.
- Example: Airbnb enables individuals to rent out their properties or spare rooms to other individuals looking for accommodations.

#### **6. Mobile Commerce (M-commerce):**

- Mobile commerce refers to transactions conducted through mobile devices such as smartphones and tablets.
- Example: The Starbucks mobile app allows users to order and pay for drinks directly from their smartphones for pickup at nearby locations.

#### **7. Social Commerce:**

- Social commerce involves the use of social media platforms to facilitate buying and selling of products and services.
- Example: Instagram Shopping allows businesses to tag products in their posts, enabling users to purchase directly from within the app.

These are just a few examples of the types of e-commerce models that exist in the real world. E-commerce continues to evolve, with new models and technologies emerging to meet the changing demands of consumers and businesses.

### **The main types of e-commerce business models include the following**

**B2B.** This refers to the electronic exchange of products, services or information between businesses rather than between businesses and consumers. Examples include online directories and exchange websites that let businesses search for products, services or information and initiate online transactions through e-procurement interfaces.

**B2C.** These transactions are when businesses sell products, services or information to consumers. There are typically intermediaries or middlemen that handle shipping, delivery and customer service, however. The term was popular during the dot-com boom of the late 1990s, when online retailers and sellers of goods were a novelty.

Today, there are innumerable virtual stores and malls on the internet selling all types of consumer goods. Amazon is the most recognized among these sites, dominating the B2C market.

**Direct-to-consumer (D2C).** This is where a business that manufactures or produces goods and services sells directly to consumers online without any middlemen or distributors involved, in contrast to B2C e-commerce.

**Consumer-to-consumer (C2C).** This is a type of e-commerce in which consumers trade products, services and information with each other online. These transactions are generally conducted through a third party that provides an online platform in which the transactions are carried out.

Online auctions and classified advertisements are two examples of C2C platforms. EBay and Craigslist are two well-known examples of these platforms. Because eBay is a business, this form of e-commerce could also be called consumer-to-business-to-consumer. Platforms like Facebook marketplace and Depop -- a fashion reselling platform -- also enable C2C transactions.

**Consumer-to-business (C2B).** This is a type of e-commerce in which consumers make their products and services available online for companies to bid on and purchase. This is the opposite of the traditional commerce model of B2C.

A popular example of a C2B platform is a market that sells royalty-free photographs, images, media and design elements, such as iStock. Another example would be a job board.

**Business-to-administration (B2A).** This refers to transactions conducted online between companies and public administration or government bodies. Many branches of government are dependent on various types of e-services or products. These products and services often pertain to legal documents, registers, Social Security, fiscal data and employment. Businesses can supply these electronically. B2A services have grown considerably in recent years as investments have been made in e-government capabilities.

**Consumer-to-administration (C2A).** This refers to transactions conducted online between consumers and public administration or government bodies. The government rarely buys products or services from individuals, but individuals frequently use electronic means in the following areas:

- **Social Security.** Distributing information and making payments.
- **Taxes.** Filing tax returns and making payments.
- **Health.** Making appointments, providing test results or information about health conditions and making health services payments.

**Mobile commerce.** Also known as m-commerce, mobile commerce refers to online sales transactions using mobile devices, such as smartphones and tablets. It includes mobile shopping, banking and payments. Mobile chatbots facilitate m-commerce, letting consumers complete transactions using voice or text conversations.

## **Advantages of E- Commerce**

The benefits of e-commerce include its availability, accessibility, speed of access, selection of goods and services and international reach.

- **Around-the-clock availability.** Aside from outages and scheduled maintenance, e-commerce sites are available 24/7, enabling visitors to browse and shop at any time. Brick-and-mortar businesses tend to open for a fixed number of hours and even close entirely on certain days.
- **Speed of access.** While shoppers in a physical store can be slowed by crowds, e-commerce sites run quickly, depending on compute and bandwidth considerations of both the consumer device and the e-commerce site. Product, shopping cart and checkout pages load in a few seconds or less. A typical e-commerce transaction requires a few clicks and takes less than five minutes.
- **Wide selection.** Amazon's first slogan was "Earth's Biggest Bookstore." It could make this claim because it was an e-commerce site and not a physical store that had to stock each book on its shelves. E-commerce enables brands to make an array of products available, which are then shipped from a warehouse or various warehouses after a purchase is made. Customers are likely to have more success finding what they want.
- **Easy accessibility.** Customers shopping in a physical store might have difficulty locating a particular product. Website visitors can browse product category pages in real time and use the site's search feature to find the product quickly.

- **International reach.** Brick-and-mortar businesses sell to customers who physically visit their stores. With e-commerce, businesses can sell to anyone who can access the web. E-commerce has the potential to extend a business's customer base.
- **Lower cost.** Pure play e-commerce businesses avoid the costs of running physical stores, such as rent, inventory and cashiers. They might incur shipping and warehouse costs, however.
- **Personalization and product recommendations.** E-commerce sites can track a visitor's browsing, search and purchase histories. They can use this data to present personalized product recommendations and obtain insights about target markets. Examples of how such insights are used include the sections of Amazon product pages labeled "Frequently bought together" and "Customers who viewed this item also viewed."

### Disadvantages of E- Commerce

The perceived disadvantages of e-commerce include sometimes limited customer service, consumers not being able to see or touch a product prior to purchase and the wait time for product shipping. Security issues can also be a problem.

- **Limited customer service.** If customers have a question or issue in a physical store, they talk to a clerk, cashier or store manager for help. In an e-commerce store, customer service can be limited. The site might only provide support during certain hours and its online service options might be difficult to navigate or not able to answer specific questions.
- **Limited product experience.** Viewing images on a webpage can provide a good sense of a product, but it's different from experiencing the product directly, such as playing a guitar, assessing the picture quality of a television or trying on a shirt or dress. E-commerce consumers can end up buying products that differ from their expectations and have to be returned. In some cases, the customer must pay to ship a returned item back to the retailer. Augmented reality is expected to improve customers' ability to examine and test e-commerce products.
- **Wait time.** In a store, customers pay for a product and go home with it. With e-commerce, customers must wait for the product to be shipped to them. Although shipping windows are decreasing as next-day and even same-day delivery becomes common, it's not instantaneous.

- **Security.** Skilled hackers can create authentic-looking websites that claim to sell well-known products. Instead, the site sends customers fake or imitation versions of those products -- or simply steals credit card information. Legitimate e-commerce sites also carry risk, especially when customers store their credit card information with the retailer to make future purchases easier. If the retailer's site is hacked, threat actors may steal that credit card information. A data breach can damage a retailer's reputation.

## **E- Commerce applications**

Many e-commerce apps use online marketing strategies to improve the customer experience and get customers to use the platform. These include email, online catalogs, shopping carts, Electronic Data Interchange (EDI), file transfer protocol, web services and mobile applications.

These approaches are used in B2C and B2B e-commerce activities, as well as other types of outreach. They include emailing targeted ads and e-newsletters to subscribers and sending text messages to mobile devices. Sending unsolicited emails and texts is generally considered spam, so more companies now try to entice consumers online, using tools such as digital coupons, social media marketing and targeted advertisements.

Another area of focus for e-commerce companies is security. Developers and admins should consider customer data privacy and security, data governance-related regulatory compliance mandates, personally identifiable information privacy rules and information protection protocols when developing e-commerce applications. Some security features are added during the design of an application, while others must be continually updated to address evolving threats and new vulnerabilities.

## **E- Commerce platforms and vendors**

An e-commerce platform is a tool that's used to manage an e-commerce business. E-commerce platform options range in size from ones for small businesses to large enterprises. They include online marketplaces, such as Amazon and eBay, that simply require signing up for user accounts and little to no IT implementation.

SaaS is another e-commerce platform model. Business owners subscribe to a service where they essentially rent space in a cloud-hosted service. This approach doesn't require in-house development or on-premises infrastructure. Other e-commerce trends include open source



platforms that require a cloud or on-premises hosting environment or complete manual implementation and maintenance.

Examples of e-commerce marketplace platforms include the following: Alibaba, Amazon, Chewy, eBay, Etsy, Newegg, Rakuten, Walmart Marketplace and Wayfair. Vendors offering e-commerce platform services for clients hosting their own online store sites include the following: Adobe Commerce, BigCommerce, Ecwid, NetSuite Commerce, Salesforce Commerce Cloud, Shopify, Squarespace and WooCommerce.

### **Government regulations for e-commerce**

In the United States, the Federal Trade Commission (FTC) and the Payment Card Industry (PCI) Security Standards Council are among the primary agencies that regulate e-commerce activities. The FTC monitors activities such as online advertising, content marketing and customer privacy. The PCI Security Standards Council develops standards and rules, including PCI Data Security Standard compliance, which outlines procedures for the proper handling and storage of consumers' financial data.

To ensure the security, privacy and effectiveness of e-commerce, businesses should authenticate business transactions, control access to resources such as webpages for registered or selected users, encrypt communications and implement security technologies, such as secure sockets layer and two-factor authentication.

### **Different E – Commerce platform**

An E - Commerce platform is a software solution that allows businesses to create and manage online stores, list products, accept payments, and handle various aspects of online sales.

### **Top E - Commerce platforms in India**

Some of the top E - Commerce platforms in India include Amazon India, Flipkart, Shopify, BigCommerce, Wix, IndiaMART, Myntra, Nykaa, Meesho, and Snapdeal.

Note: To choose the right E- Commerce platform, consider factors such as your business size, budget, product range, customization needs, and target audience. Compare the features and pricing of different platforms to make an informed decision.

## **The three main types of E-Commerce platforms are**

1. SaaS (software-as-a-service) platforms
2. PaaS (platform-as-a-service) platforms
3. On-premises platforms

SaaS and PaaS platforms both deliver the e-commerce solutions through the internet. SaaS platforms, like Shopify, involve software only. Once they add a hardware element as well, they become known as PaaS platforms.

On-premises platforms are hosted locally by the retailer and managed by their IT staff rather than being set up by another provider and accessed through the cloud.

SaaS and PaaS options are best for smaller companies or companies that are just getting started in e-commerce. They offer professional setup and support, but they usually charge a monthly fee to use the site in addition to transaction fees on every purchase. On-premise platforms allow a company to have more control over its e-commerce site and create a custom storefront solution.

There are numerous e-commerce platforms available presently, each offering different features, pricing structures, and target audiences. Here are some of the most popular ones along with examples and explanations:

1. Shopify:
  - Shopify is a comprehensive e-commerce platform that allows businesses to create and customize their online stores easily.
  - It offers a wide range of features including website design tools, payment processing, inventory management, and marketing tools.
  - Example: Allbirds, a sustainable footwear company, uses Shopify to power its online store, offering a seamless shopping experience for customers.
2. WooCommerce:
  - WooCommerce is an open-source e-commerce plugin designed for WordPress websites, allowing users to turn their WordPress sites into online stores.
  - It offers flexibility and customization options through various themes and plugins.
  - Example: Beardbrand, a men's grooming products company, uses WooCommerce to sell its products directly from its WordPress website.

### 3. Magento:

- Magento is a powerful e-commerce platform known for its scalability and customization capabilities, making it suitable for businesses of all sizes.
- It offers advanced features such as multi-store management, flexible product cataloging, and extensive third-party integrations.
- Example: Nike uses Magento to manage its online store, providing a seamless shopping experience for customers worldwide.

### 4. BigCommerce:

- BigCommerce is a cloud-based e-commerce platform that caters to businesses looking for scalability and robust features.
- It offers built-in marketing tools, customizable themes, and integrations with popular third-party applications.
- Example: Skullcandy, a headphone and earbud manufacturer, uses BigCommerce to power its online store, offering a user-friendly shopping experience for its customers.

### 5. Wix:

- Wix is a website builder platform that also offers e-commerce capabilities, allowing users to create online stores with drag-and-drop tools.
- It offers a range of templates, payment options, and marketing tools to help businesses create professional-looking online stores.
- Example: MVMT, a watch and accessories brand, uses Wix to build and manage its online store, providing a visually appealing shopping experience for its customers.

### 6. Squarespace:

- Squarespace is a website builder platform that includes e-commerce functionality, enabling users to create online stores with ease.
- It offers beautifully designed templates, integrated marketing tools, and flexible product display options.
- Example: M.Gemi, a luxury footwear brand, uses Squarespace to power its online store, offering a seamless shopping experience for customers.

These are just a few examples of e-commerce platforms available in the market. Each platform has its own set of features and benefits, so businesses should carefully evaluate their needs and choose the platform that best suits their requirements.

## Top websites developed using Magento

Magento is a platform with built-in PHP, which helps programmers create E-Commerce websites. It was released on March 31, 2008, by Varien and developed on Zend Framework. Up to this time, there are hundreds of thousands of businesses using this open-source to start their online store including big ones like Nike, Samsung, etc.

In 2018, Adobe acquired Magento with a price of \$1.68 billion to complete their commerce loop. This acquisition will also help Magento develop in the enterprise market.

### Examples of top websites developed using Magento:

#### **Sigma Beauty (sigma.com):**

Sigma Beauty is a cosmetics brand known for its high-quality makeup brushes and beauty products.

The website, built on Magento, offers a user-friendly shopping experience with easy navigation, product categorization, and secure checkout.

#### **Sanctuary Clothing (sanctuaryclothing.com):**

Sanctuary Clothing is a women's fashion brand offering contemporary apparel and accessories.

Their Magento-based website features a sleek design, seamless product browsing, and personalized shopping experiences.

#### **Högl Shoes (hogl.com):**

Högl is an Austrian shoe brand specializing in high-quality footwear for women.

Their Magento website provides an elegant and intuitive interface for customers to explore their range of shoes and make purchases.

#### **Harvey Nichols (harveynichols.com):**

Harvey Nichols is a luxury department store offering fashion, beauty, food, and wine.

Their Magento-powered website delivers a premium shopping experience with sophisticated design, rich product imagery, and seamless checkout processes.

#### **Forever New (forevernew.com.au):**

Forever New is an Australian fashion retailer known for its trendy clothing and accessories for women.

Their Magento-based website offers a visually appealing layout, personalized recommendations, and convenient shopping features.

**Kipling (kipling.com):**

Kipling is a global fashion brand famous for its stylish and functional bags and accessories.

Their Magento website provides a user-friendly interface with easy navigation, detailed product descriptions, and secure payment options.

**Osprey London (ospreylondon.com):**

Osprey London is a British luxury leather goods brand offering handbags, accessories, and gifts.

Their Magento-powered website showcases their exquisite product range with high-resolution images, detailed product descriptions, and a seamless shopping journey.

**Lards (lards.co.uk):**

Lards is a gourmet food and gift retailer offering a wide range of artisanal products.

Their Magento website features a clean and organized layout, making it easy for customers to browse and purchase their products.

**Wall Art Prints (wallartprints.com.au):**

Wall Art Prints specializes in high-quality canvas prints and custom wall art.

Their Magento-based website offers a user-friendly interface with intuitive search functionality, customizable options, and secure checkout.

**Warby Parker (warbyparker.com):**

Warby Parker is a popular eyewear brand known for its affordable and stylish glasses.

Their Magento website provides a seamless shopping experience with virtual try-on tools, easy prescription uploads, and home try-on services.

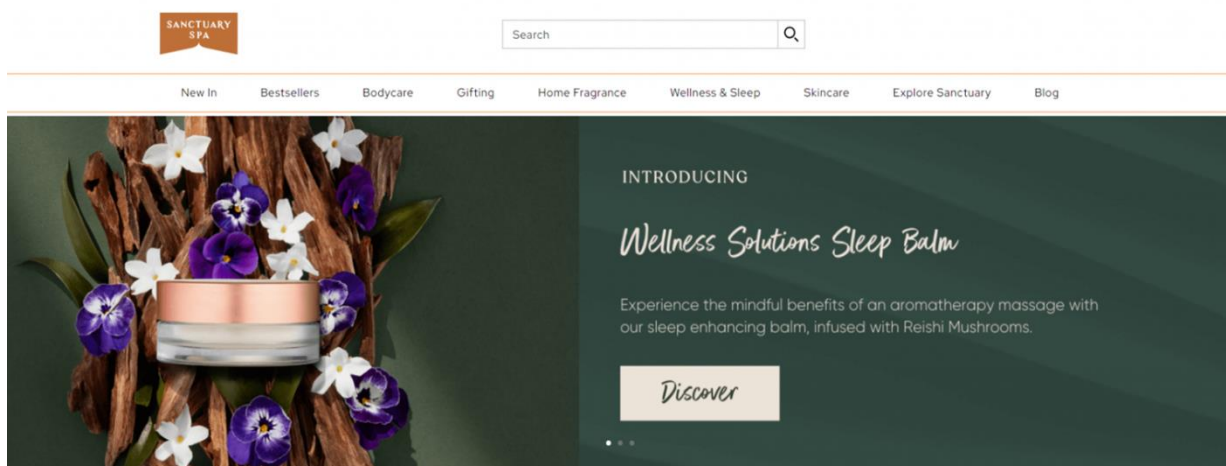
These websites demonstrate the versatility and scalability of Magento as an e-commerce platform, catering to various industries and providing engaging online shopping experiences for customers.

**Example:**

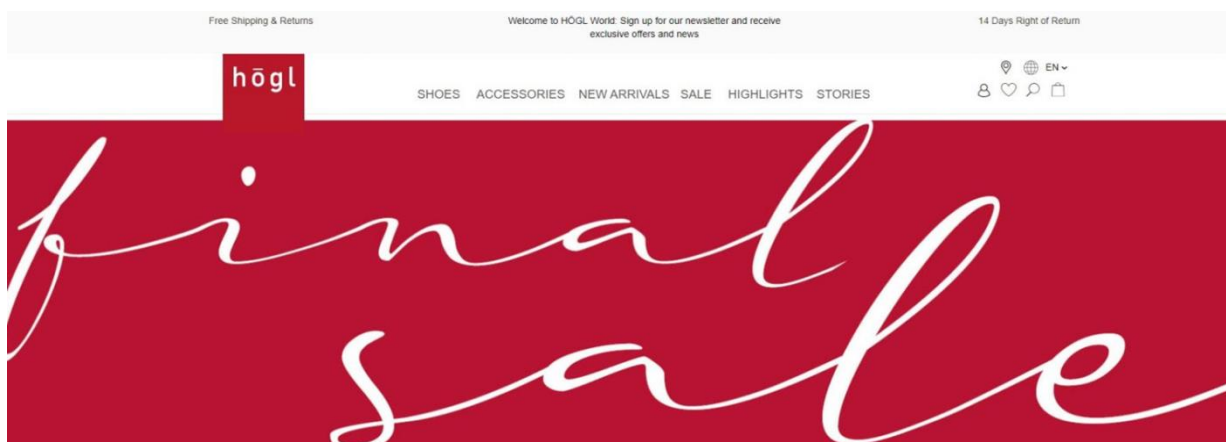
**Sigma Beauty (sigma.com):**



**Sanctuary Clothing (sanctuaryclothing.com):**



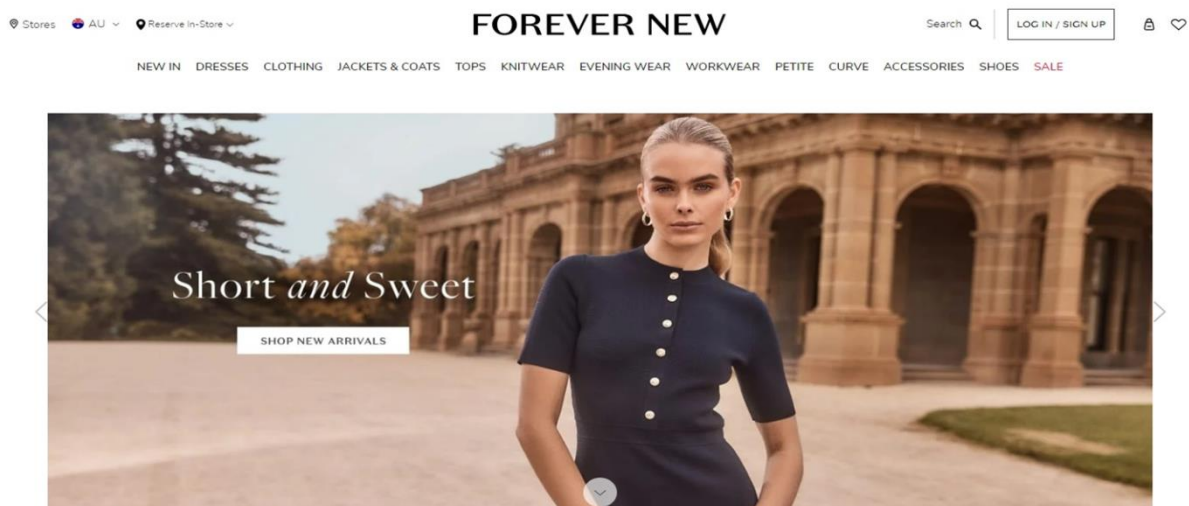
**Högl Shoes (hogl.com):**



**Harvey Nichols (harveynichols.com):**

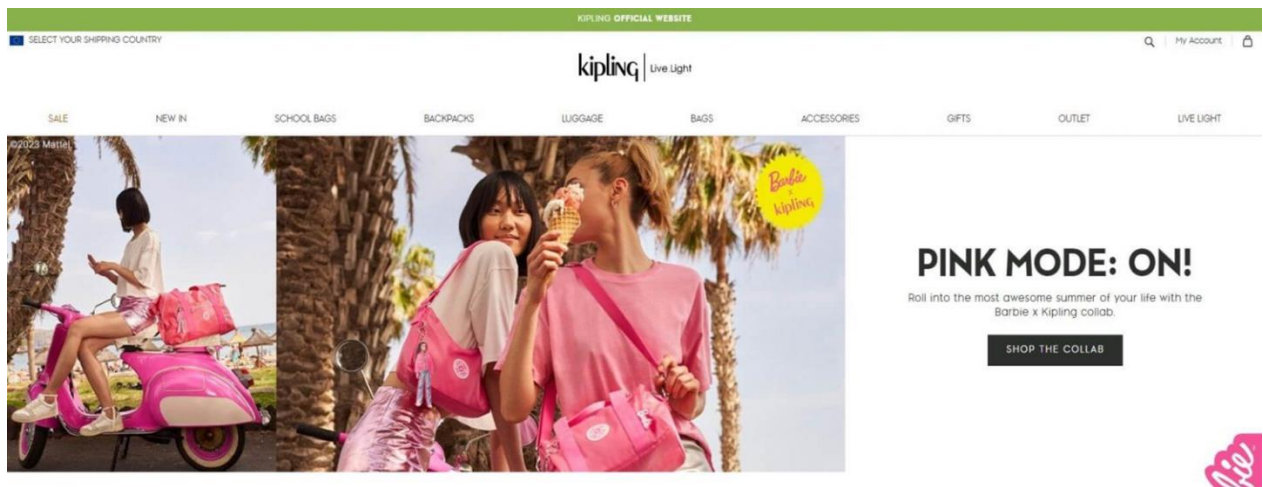


**Forever New (forevernew.com.au):**





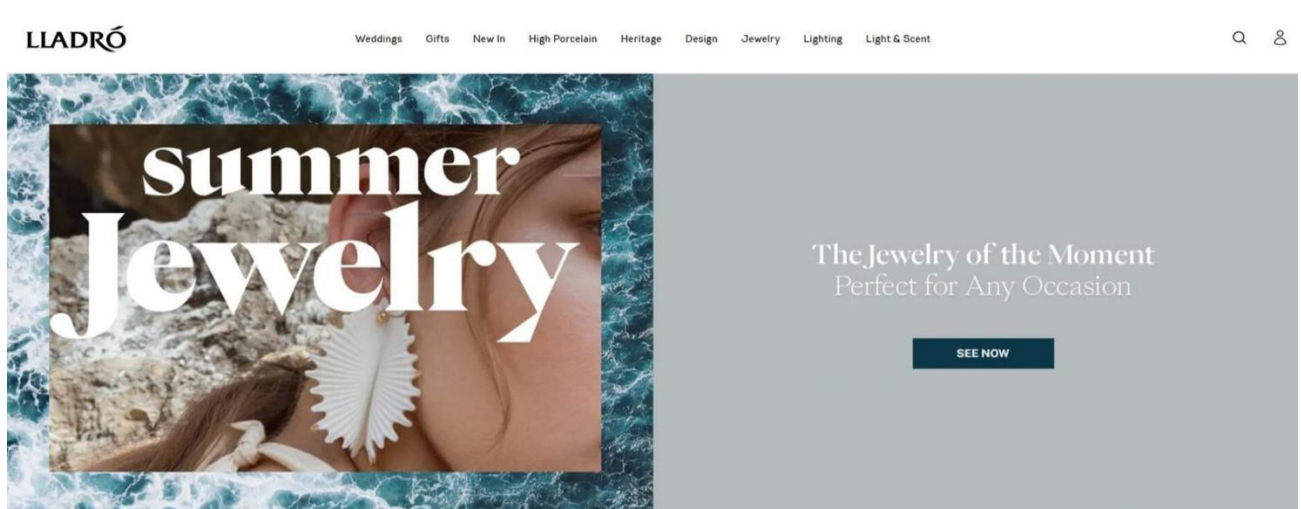
## Kipling (kipling.com):



## Osprey London (ospreylondon.com):

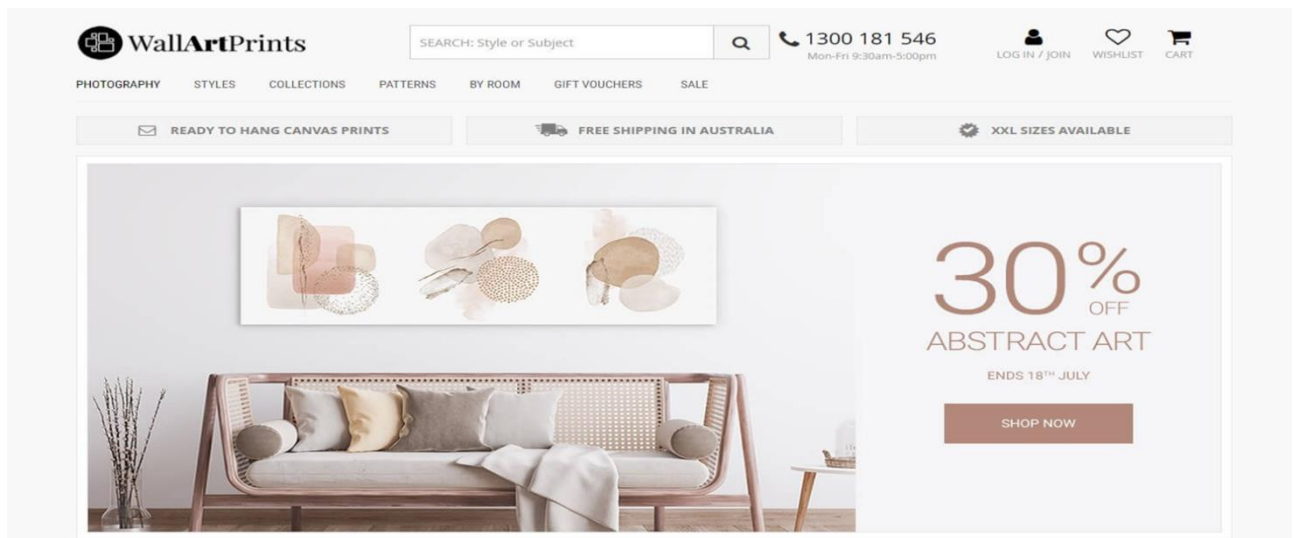


## Lards (lards.co.uk):

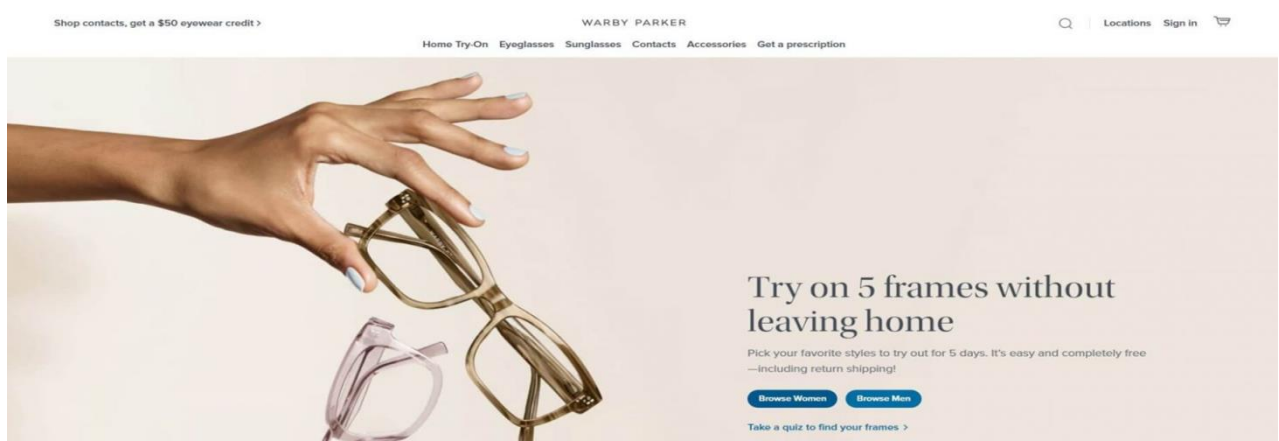




## Wall Art Prints (wallartprints.com.au):



## Warby Parker (warbyparker.com):



## What is Magento?

Magento is an E Commerce platform built on open source technology which provides online merchants with a flexible shopping cart system, as well as control over the look, content and functionality of their online store. Magento offers powerful marketing, search engine optimization, and catalog-management tools. Magento is a platform with built-in PHP, which helps programmers create E Commerce websites. It was released on March 31, 2008, by Varien and developed on Zend Framework. Up to this time, there are hundreds of thousands of businesses using this open-source to start their online store including big ones like Nike, Samsung, etc.

## Why to Use Magento?

For businesses and individuals who want to develop a long-term professional E Commerce site, should think of Magento as a priority. Magento is the best open-source to build an online store. But the vital thing is it's very professional and totally free. Customers need to understand a little English and Technology is absolutely able to build them a "monumental" site with this open source.

## There are some basic features of Magento

1. **Product management:** with many images, optional comment reviews on products, a favorites list, and inventory.
2. **Category management:** easy to find and select products by category
3. **Inventory management:** management of products left in stock, import, export ...
4. **Client account:** account status, transaction history, preferred catalog, address, shopping cart ...
5. **Customer service:** enhance the features, customer contact form; comprehensive follow-up, and email service.
6. **Order Management**
7. **Payments:** Many payment methods such as credit card, PayPal, Authorize.net, Google Checkout, support for external payment modules like CyberSource, ePay, eWAY, and more.
8. **Search technology:** fast, friendly, supports search on Google Sitemap
9. **International support:** multi-lingual and monetary
10. **Promotional and marketing tools:** coupons, promotions, and more.
11. **Analyze and report:** integrate with the Google Analytics service and provide multiple reports.

Magento is a highly popular and robust eCommerce platform that empowers thousands of retailers and brands to create and manage their online stores easily. As an open-source solution, Magento 2 offers a wide range of benefits and features that make it the go-to choice for many businesses. You get a range of basic features to enhance your online store:

## Online Store

Product Management: Easily manage products by adding multiple images, allowing customers to leave optional comment reviews, creating a favorites list, and keeping track of inventory.

Category Management: Organize products into categories, making it simple for customers to find and select desired items.

Inventory Management: Keep track of product stock levels, import and export data, and efficiently manage inventory.

Client Accounts: Provide customers with their accounts, granting access to account status, transaction history, preferred catalogs, address details, and shopping cart contents.

Customer Service: Improve customer service with features like customer contact forms, comprehensive follow-up options, and automatic email services, ensuring effective communication.

Order Management: Streamline the order process by efficiently managing, tracking, and fulfilling orders.

Magento Payments: Enable various payment methods, including credit cards, PayPal, Authorize. net, and Google Checkout, and support external payment modules.

Magento Live Search Technology: Implement fast and user-friendly search features, including Google SiteMap integration, to help customers find products quickly.

International Support: Facilitate international sales with multi-lingual and multi-currency support, enabling seamless global transactions.

Promotional and marketing tools: Boost sales and engagement with marketing tools such as coupons, discounts, and other marketing features.

Analyze and Report: Integrate with Google Analytics to gather valuable insights and generate detailed reports on your online store's performance.

## **Magento Version and Features**

Magento has undergone many changes since its release in 2008. Each new Magento version has introduced features and improvements to enhance the online shopping experience for customers and merchants. There are two main Magento versions: Magento 1 and Magento 2. Magento 1 was released in 2008 and reached end of life in 2020. Its successor, Magento 2, was launched in 2018 with advanced security features, robust API support, and future-proof functionality. Despite Magento 1 reaching its end of life, more than 27,000 websites still use Magento 1. If you're one of them, you should consider migrating from Magento 1 to Magento 2 or using Nexcess Safe Harbor to protect your Magento 1 store.

Every new Magento version builds on the success of the previous one with new features. Here's a Magento versions list that compares major releases and highlights key changes to the platform since its initial release.

Magento version release	Release dates	Key features added	End of support
Magento 1.0-1.9x	March 31, 2008–2015.	The first stable Magento versions with essential ecommerce functionality.	June 30, 2020.
Magento 2.0	November 17, 2015–February 27, 2018.	A total transformation of the Magento platform. Significantly faster, more secure, and user-friendly.	March 31, 2018.
Magento 2.1	June 23, 2016–June 25, 2019.	Content staging and preview, Solr replaced with Elasticsearch for Commerce versions, PayPal payment methods, and improved admin interface.	June 30, 2019.
Magento 2.2	September 26, 2017–January 28, 2020.	Advanced reporting functionality, instant purchase checkout, Magento shipping, and new B2B features.	December 1, 2019.
Magento 2.3	November 28, 2018–October 12, 2021.	Progressive web apps, multi-source inventory, Elasticsearch introduced for community versions, and improved page builder.	September 8, 2022.
Magento 2.4.0–2.4.3	July 28, 2020–October 12, 2021.	2FA for the admin panel, enhanced media gallery, MySQL search engine fully replaced by Elasticsearch, seller-assisted shopping, and improved progressive web apps.	November, 2022.
Magento 2.4.4–2.4.6 (current Magento version)	April 12, 2022–March 14, 2023.	Support for OpenSearch, vendor bundled extensions removed, improved page builder, cart and checkout updates, and new payment methods.	

Table -1 Magento Version and Features

## New Version Magneto

Magento 2.4.6 is the latest Magento version. Released on March 14, 2023, it contains over 300 quality fixes and upgrades that improve Magento performance, security, and scalability.

The latest Magento version, 2.4.6, adds support for PHP 8.2 and removes support for PHP 7.4, which reached end of life on November 28, 2022. It also contains GraphQL operation optimizations for category tree rendering response and bulk cart operations.

### Magento version 2.4.6: key updates

Example: Key improvements in Magento 2.4.6 include:

Security enhancements: Magento 2.4.6 provides security updates and improvements, such as checkout reCAPTCHA validation fixes and a new system configuration that requires email confirmation when an admin user changes their email.

Platform enhancements: The latest Magento version supports PHP 8.2, Composer 2.2.x, Redis 7.0.x, Elasticsearch 8.0.x, and MariaDB 10.6 (LTS). It also supports OpenSearch v2.x as the default search engine for Magento 2.

Performance and scalability enhancements: Magento 2.4.6 introduces a new setting that lets you limit the number of products displayed in the product grid to improve performance.

Custom SMTP: Magento 2.4.6 lets you configure a custom SMTP provider from the admin panel. You no longer need a third-party extension to send emails from your Magento store.

## **Advantage and Disadvantage of Magento versions**

Magento 2 is quite different from its previous versions. Magento 2 is a user-friendly. C-commerce platform. If we want to work with Magento 2, then we should also be aware of its advantages and disadvantages. It has advantages along with some disadvantages, which are given below.

### **Advantage**

1. Category and product creation is much easier and quicker than the other.
2. It offers various templates and theme that allows both the Magento designer and store owner to customize the look and feel and even also optimize it for the Smartphone.
3. Magento 2 offers amazing Admin interface, which is very simplified. Non-technical admin likes this simplified interface.
4. LESS preprocessor and CSS URL helps to increase performance, speed up the page load, and also make system friendly with the search engine.
5. It is developer-friendly, as it built according to the modern development logic that makes it more efficient for the developer to work with it.
6. Magento 2 has Luma theme, which is an eye-catching theme and easy to learn. Using the Luma theme, we can easily manage our online store in very less time. It also helps us to enhance the productivity of creating products, managing orders, and customer's data.

### **Disadvantage**

1. All the extensions of Magento 1.x will not work on Magento 2.0. You need to purchase the extensions and integrate them into Magento 2.
2. Magento 2 Community Edition is free, whereas Magento 2 Enterprise Edition is paid. You have to pay for Magento 2 Enterprise Edition.
3. Development expertise is required due to its complexity.
4. Magento 1 theme "Madison Island" cannot be transferred to the Magento 2, it will need to be built from scratch, which makes it an expensive endeavor.
5. For the small store, it is a little pricey. The enterprise edition is expensive of the Magento 2.
6. Magento 2 needs heavy servers, which makes it slow and requires intensive data input for optimal performance.

## Installation Steps for Magento:

### There are two ways we can install magento2

**Using Wamp Server:** WampServer refers to a software stack for the Microsoft Windows operating system, created by Romain Bourdon and consisting of the Apache web server, OpenSSL for SSL support, MySQL database and PHP programming language.

**Using Xampp Server:** XAMPP is an abbreviation for cross-platform, Apache, MySQL, PHP, and Perl, and it allows you to build Magento site offline, on a local web server on your computer. This simple and lightweight solution works on Windows, Linux, and Mac – hence the “cross-platform” part.

So, to install Magento2 on our machine, we need to first install Xampp or Wamp Server.

Step 1: Install Apache2 PHP and Required Extensions. Step 1.1 Install

Apache2 Server.

Step 2: Install Database Server.

Step 3: Create MySQL User (Required)

Step 4: Install Composer.

Step 5: Download Magento 2 Pack.

Step 6: Install Magento 2.

## System Requirements for in installation

### Example 1 : Installation Requirements for Magento 2.4.6:

#### Step 1: Operating systems

Distributions of Linux, including RedHat Enterprise Linux (RHEL), CentOS, Ubuntu, Debian, macOS, and Windows.

#### Step 2: Memory requirement

Magento2 requires 4 GB or higher RAM.

#### Step 3: Composer

Composer 2.x will be supported by Magento.

#### Step 4: Web servers

Apache 2.4

#### Step 5: Database

MySQL 8.0

MariaDB 10.4

#### Step 6: PHP

Magento 2.4.6 supports PHP 8.2

#### Step 7: Elasticsearch

As of 2.4.6 Magento, MySQL is no longer available for search purposes. You're supposed to use Elasticsearch. Elasticsearch 2.x, 5.x, and 6.x are no longer supported by Magento.

Note: To get detailed information about system requirements, just visit the mentioned link: <https://devdocs.magento.com/guides/v2.4/install-gde/system-requirements.html>

**Example 2:**

- **OS:** Windows 10
- **SERVER:** Apache 2
- **RAM:** 4 GB
- **PHP:** 8.2
- **MariaDB:** 10.4
- **Composer:** 2.6.3
- **Elasticsearch:** 7.17.0

1. **Windows 10** is a Microsoft operating system for personal computers, tablets, embedded devices and internet of things devices.

2. **As a Web server**, Apache is responsible for accepting directory (HTTP) requests from Internet users and sending them their desired information in the form of files and Web pages.

3. The **4 gb ram means** the amount of memory to work with the apps and programs, 4 gb is almost the top of the line.

4. a major update of **the PHP language**. It contains many new features and optimizations including named arguments, union types, attributes, constructor property promotion, match expression, nullsafe operator, JIT, and improvements in the type system, error handling, and consistency.

5. **MariaDB is a database.** MariaDB is very similar to MySQL (a database management system) and, in fact, a fork to MySQL. The MariaDB database is used for various purposes such as data warehousing, e-commerce, enterprise-level features, and logging applications

6. **Composer** is an application-level dependency manager for the PHP programming language that provides a standard format for managing dependencies of PHP software and required libraries

7. Magento **Elasticsearch** is a store-level search engine that delivers store-wide search results so that visitors can quickly see the list of products they wish to buy. Magento Elasticsearch is a fantastic tool for offering related products and deploying upsell and cross-sell tactics for store owners. Elasticsearch is a distributed search and analytics engine built on Apache Lucene

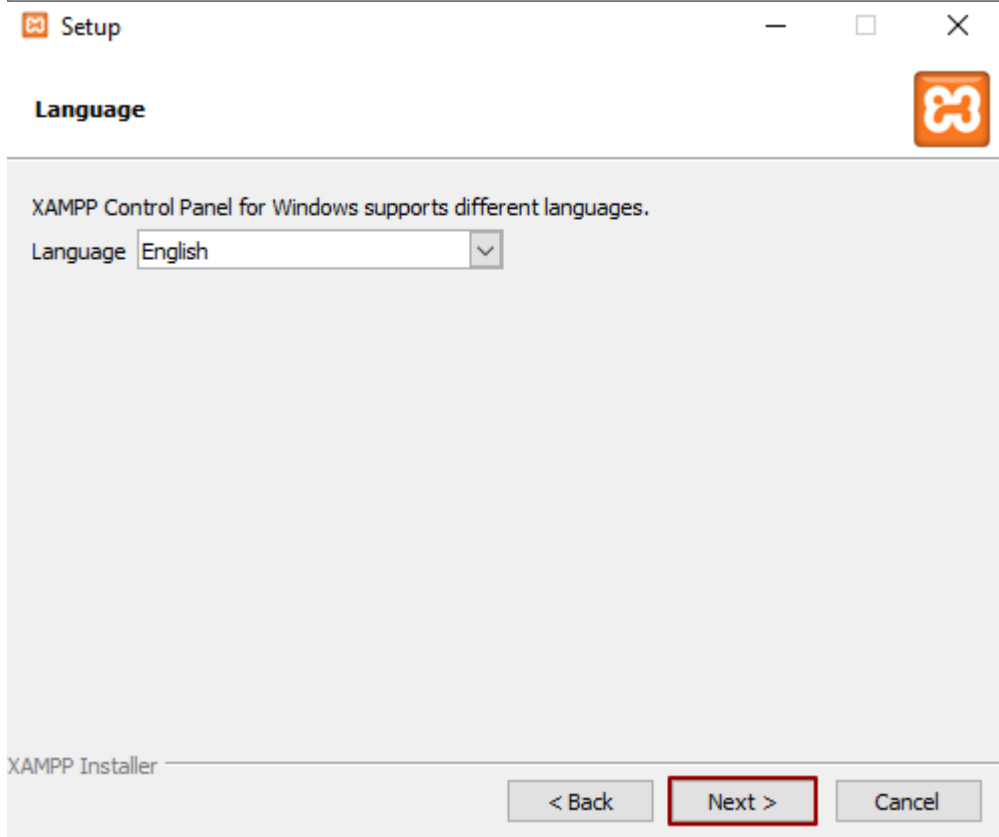
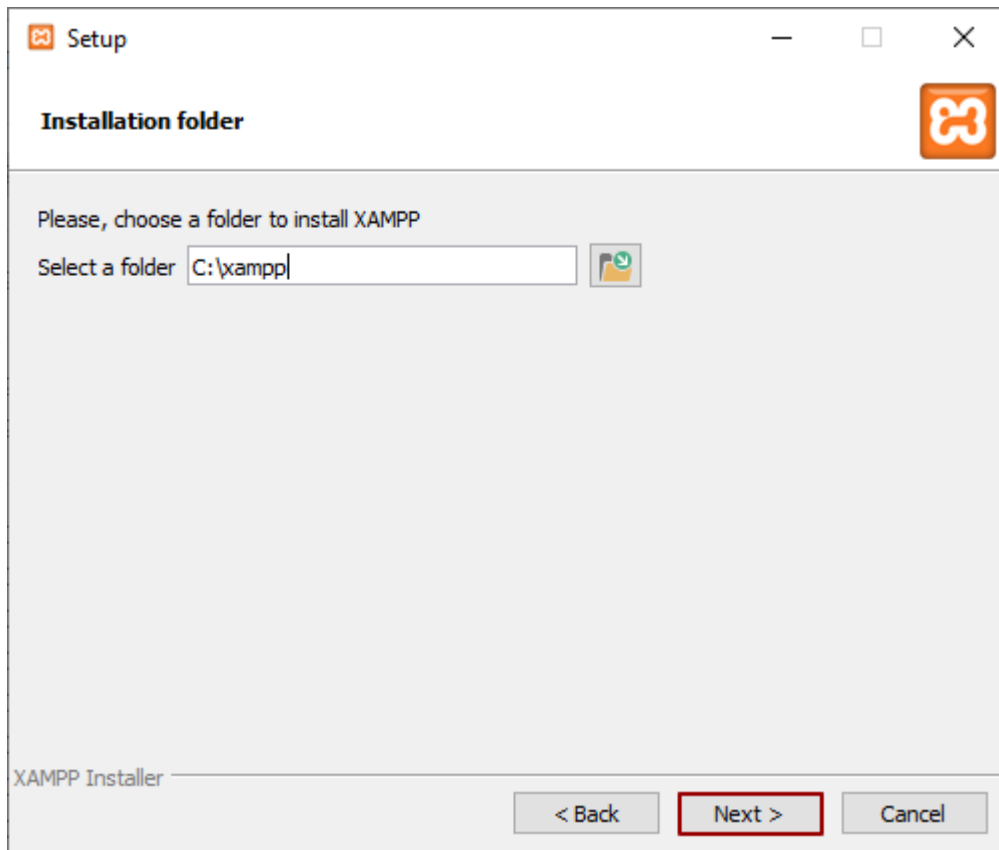
## Install Magento 2 on windows platform:

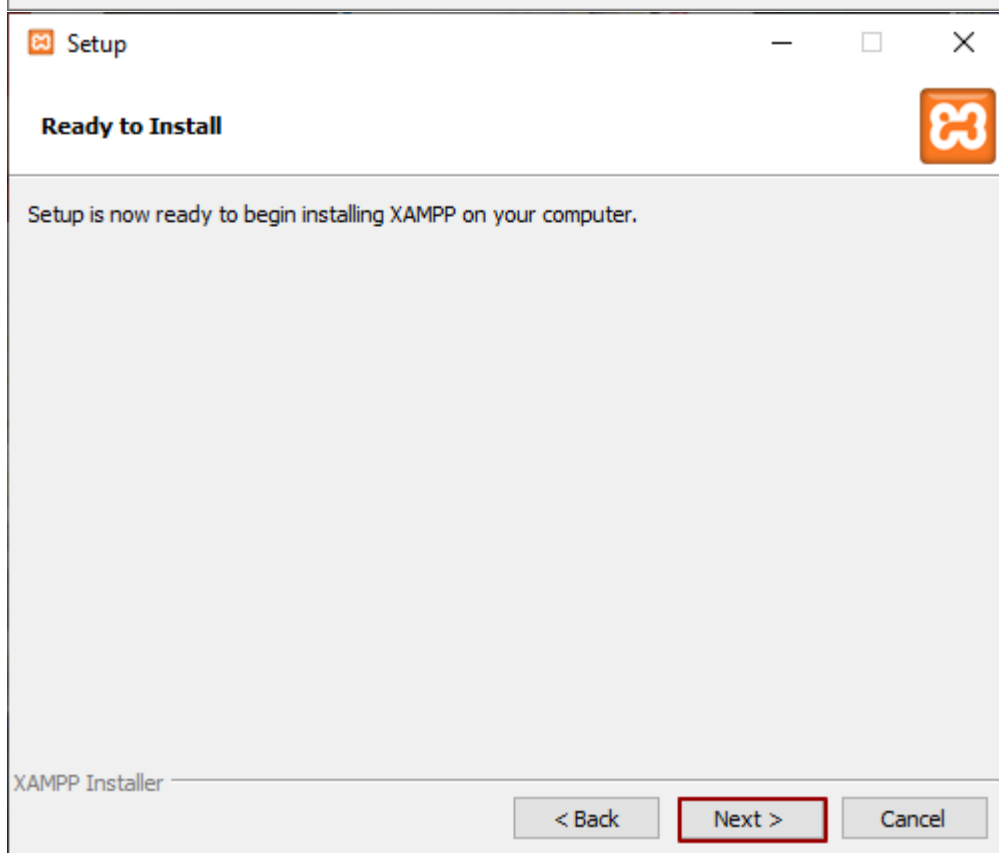
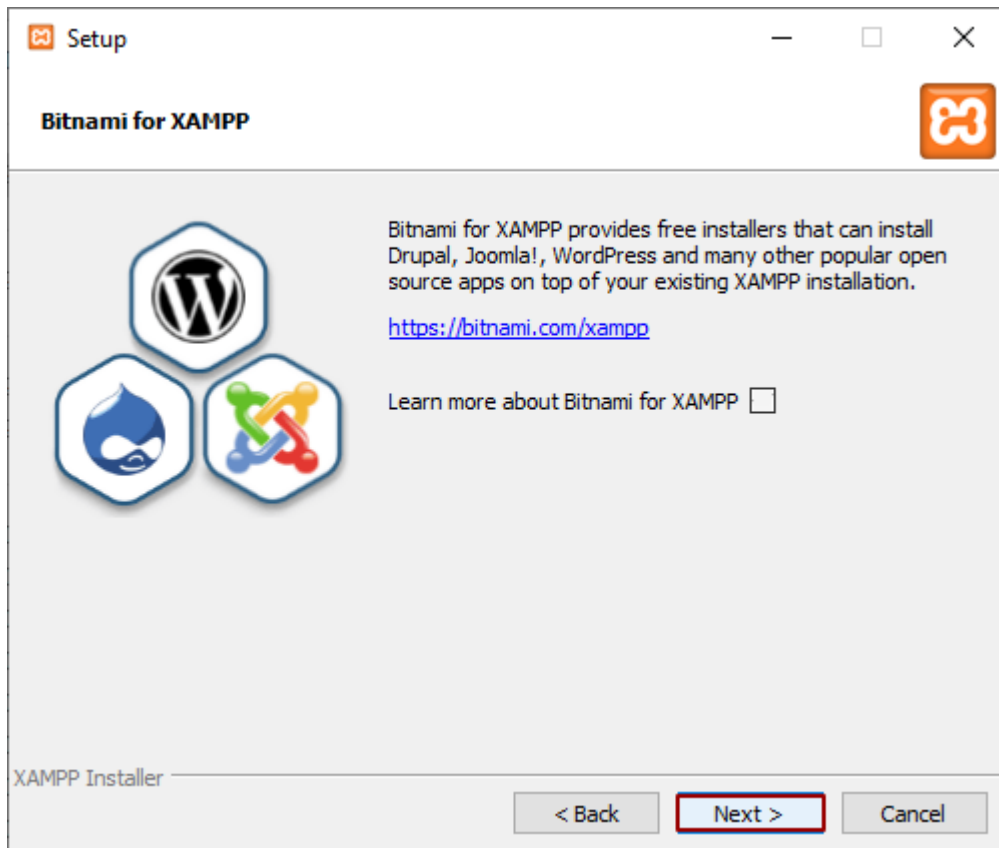
### STEP 1: Download And Configure The XAMPP Web Server.

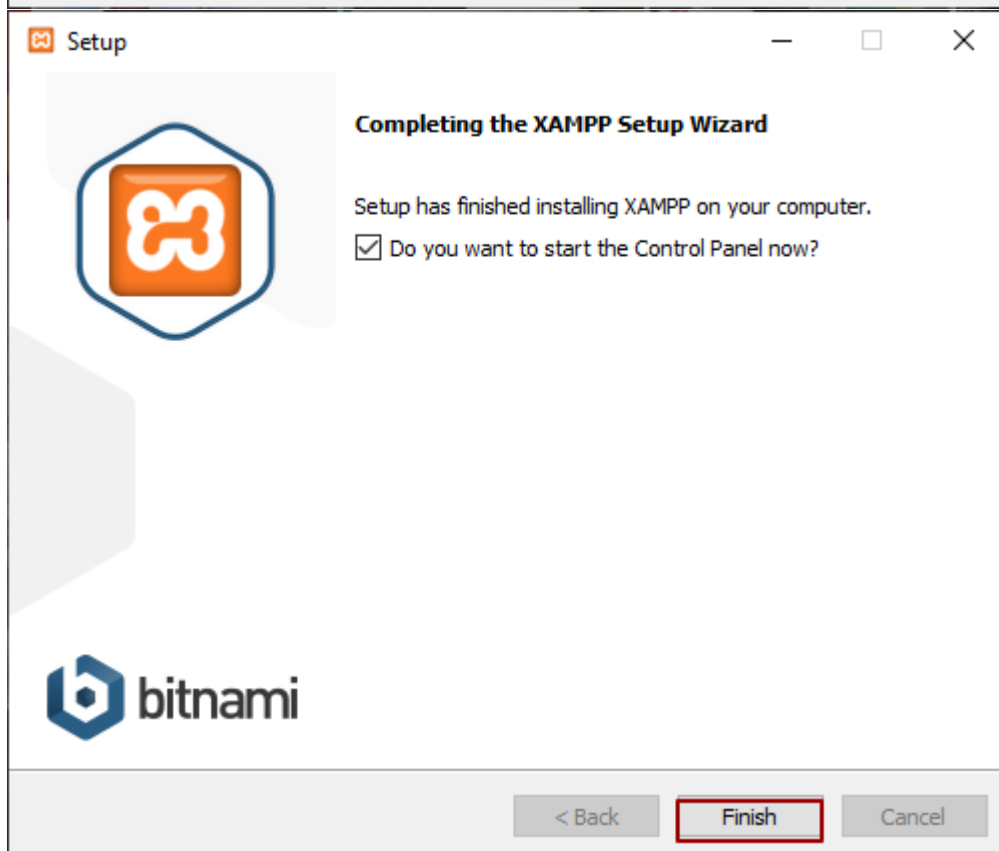
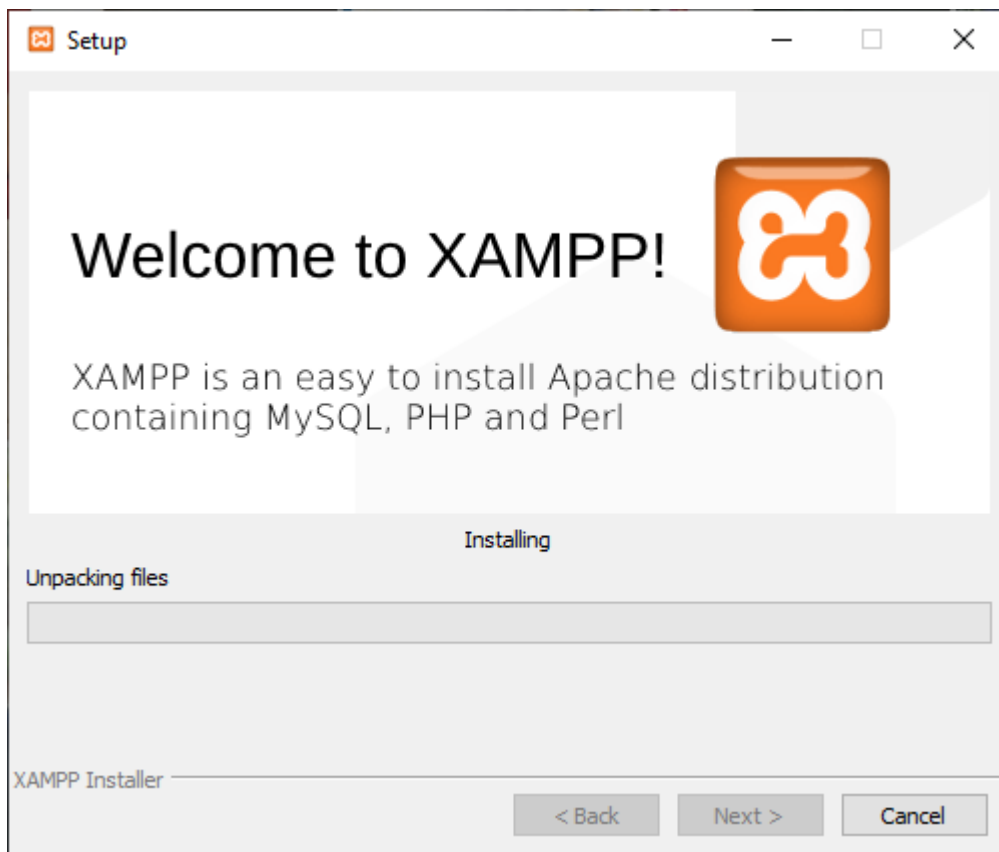
Firstly, download the XAMPP Web Server version 8.2.4 to set up the environment for Magento2 Community Edition. After that, tap “Next” in every setup window and Install the XAMPP Web Server.



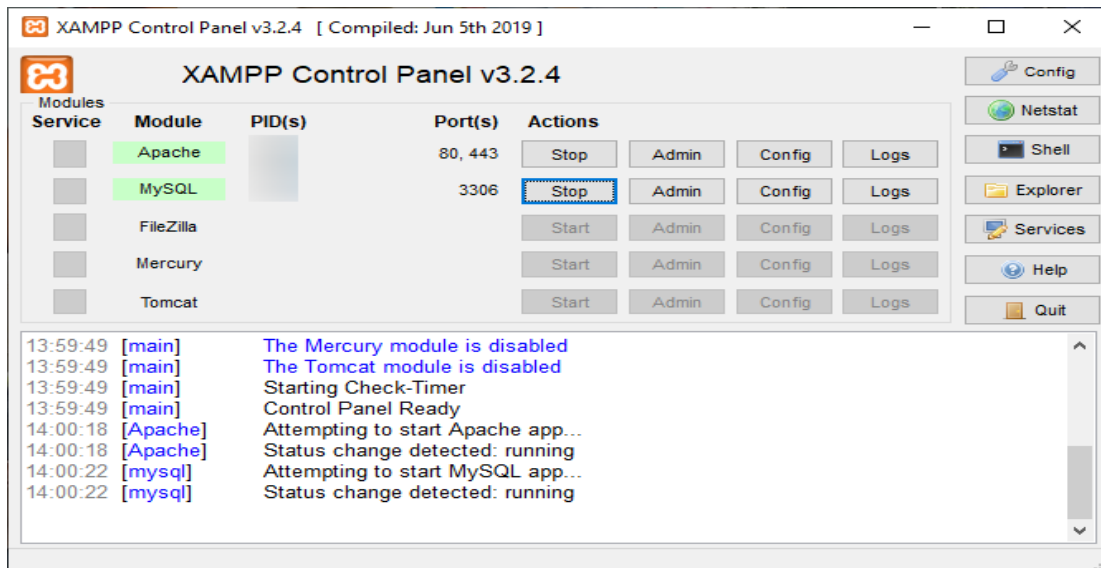




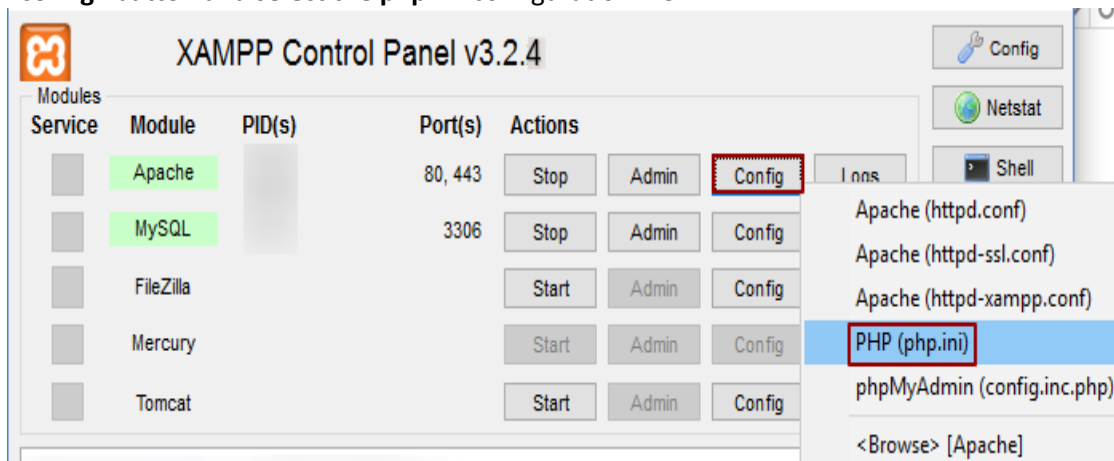




After successful installation, run your XAMPP Web Environment and Launch Apache and MySQL Services.

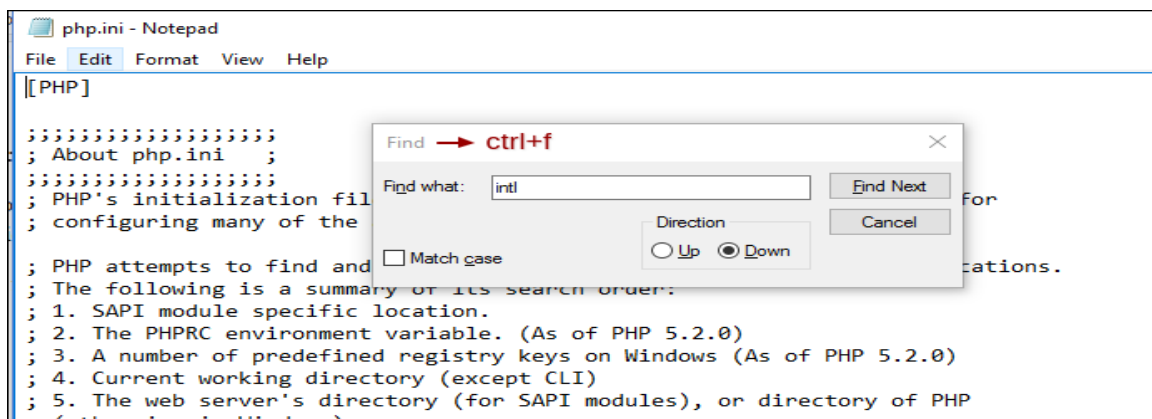


Before proceeding further, you need to allow the requisite PHP extensions. So, **just tap the apache “config” button** and **select the php.ini** configuration file.



**Enable “intl”, “xsl”, “soap”, “gd”, “sodium”, “zip”, and “sockets” extensions** in the configuration file by searching for the extensions by **pressing ‘ctrl+f’** as shown in the below images. Then remove the **semicolon to uncomment** it before the line.

**Note:** By default, the rest of the PHP extensions are enabled. If it’s not then you can enable it now or even later by following the above way. In case, if you get any error defining the necessity of any other PHP extension.



```

;extension=gmp
extension=intl
;extension=imap
;extension=xmlrpc
extension=xsl

; The MIBS data available in the PHP distribution must be installed.
; See http://www.php.net/manual/en/snmp.installation.php
;extension=snmp

extension=soap
;extension=sockets

```

Now, Look for “**extension=**” by pressing ‘**ctrl=f**’. you will find this below:

```

;;;;;;;;;;;;;;;;;;;;;;;;;
; Dynamic Extensions ;
;;;;;;;;;;;;;;;;;;;;;;;;;

; If you wish to have an extension loaded automatically, use the following
; syntax:
;
; extension=modulename
;

```

after that, replace the “**modulename**” with “**php\_Sockets.dll**” and uncomment it by removing the semi-colon like below.

```

;;;;;;;;;;;;;;;;;;;;;;;;;
; Dynamic Extensions ;
;;;;;;;;;;;;;;;;;;;;;;;;;

; If you wish to have an extension loaded automatically, use the following
; syntax:
;
extension=php_sockets.dll
;

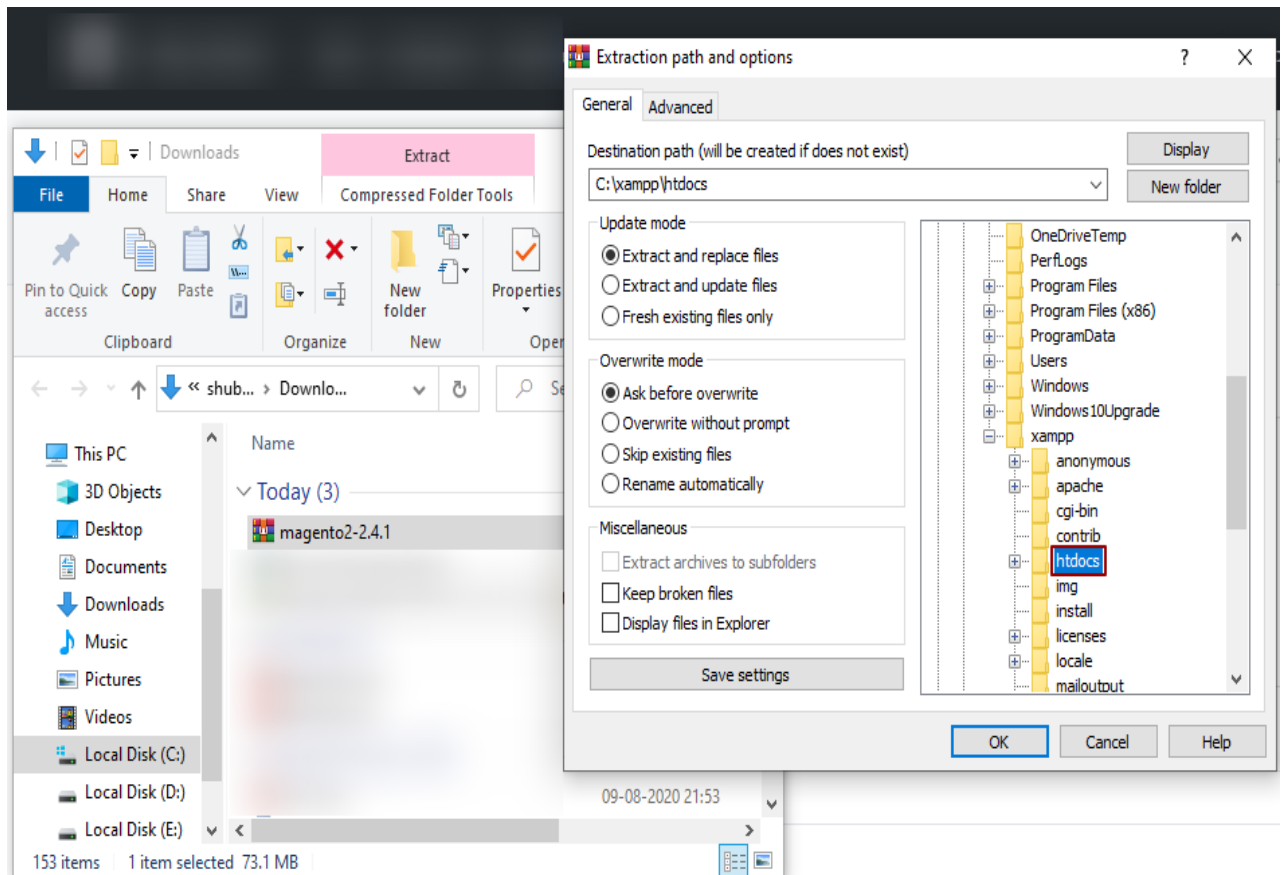
```

Save and close the file at last. Stop and restart the Apache.

## STEP 2: Download Magento 2 By Using The Composer.

Upon completion, download the Magento 2 Community Edition zip file from the mentioned link: <https://github.com/magento/magento2/releases>

**Extract the Zip file in the htdocs folder of the XAMPP installation directory** after it has been downloaded. To do that, right-click the **Magento2 zip file** and select extract files, and then select the **htdocs destination** path and press OK.



Now go to the folder htdocs and find the extracted Magento folder. **Rename this folder to Magento2.**

### STEP 3: Now, Open The “Gd2.Php” File Located As Per The Following Path:

magento2\lib\internal\magento\framework\image\adaptor\Gd2.php  
find the below lines;

```

Gd2.php - Notepad
File Edit Format View Help
$this->_fileName
);
}

/**
 * Checks for invalid URL schema if it exists
 *
 * @param string $filename
 * @return bool
 */
private function validateURLScheme(string $filename) : bool
{
    $allowed_schemes = ['ftp', 'ftps', 'http', 'https'];
    $url = parse_url($filename);
    if ($url && isset($url['scheme']) && !in_array($url['scheme'], $allowed_schemes)) {
        return false;
    }

    return true;
}

/**
 * Checks whether memory limit is reached.
 *

```

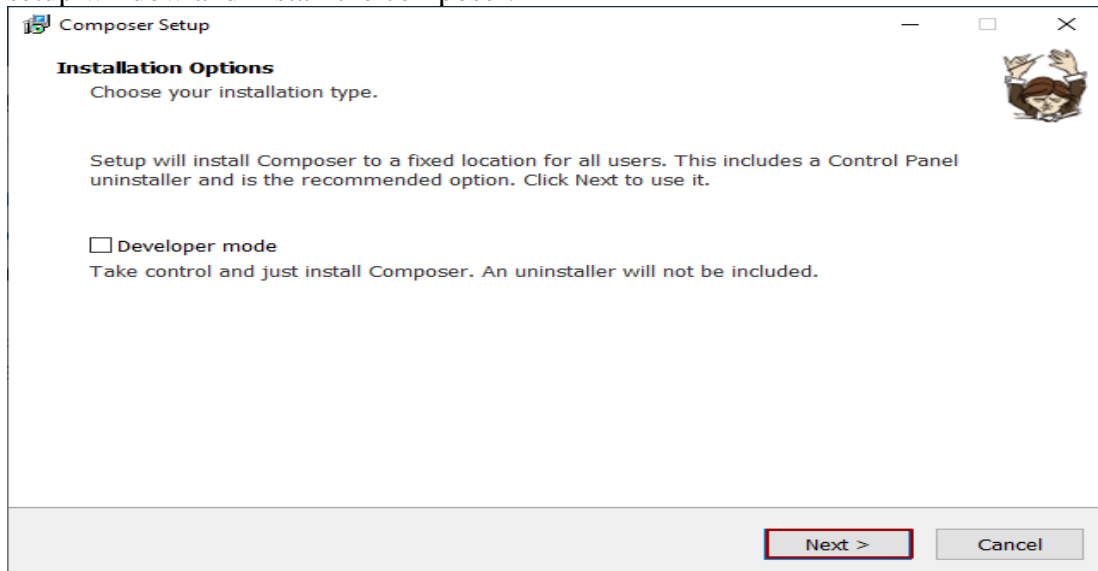
and then replace it with,

```
$allowed_schemes = ['ftp', 'ftps', 'http', 'https'];  
  
$url = parse_url($filename);  
  
if ($url && isset($url['scheme']) && strlen($url['scheme']) > 1 &&  
!in_array($url['scheme'], $allowed_schemes)) {  
  
return false;
```

and at last, **save the file** and exit.

#### **STEP 4: Download And Install Composer:**

On your Windows system download Composer by using the link: <https://getcomposer.org/Composer-Setup.exe> and then install it. Tap “Next” in every setup window and Install the composer.



The **php.exe** file path must be defined next. Provide the path and then click “Next”

Composer Setup

Settings Check

We need to check your PHP and other settings.

Choose the command-line PHP you want to use:

C:\xampp\php\php.exe

Browse...

☒ Add this PHP to your path?

< Back

Next >

Cancel

Composer Setup

Proxy Settings

Choose if you need to use a proxy.

☐ Use a proxy server to connect to internet

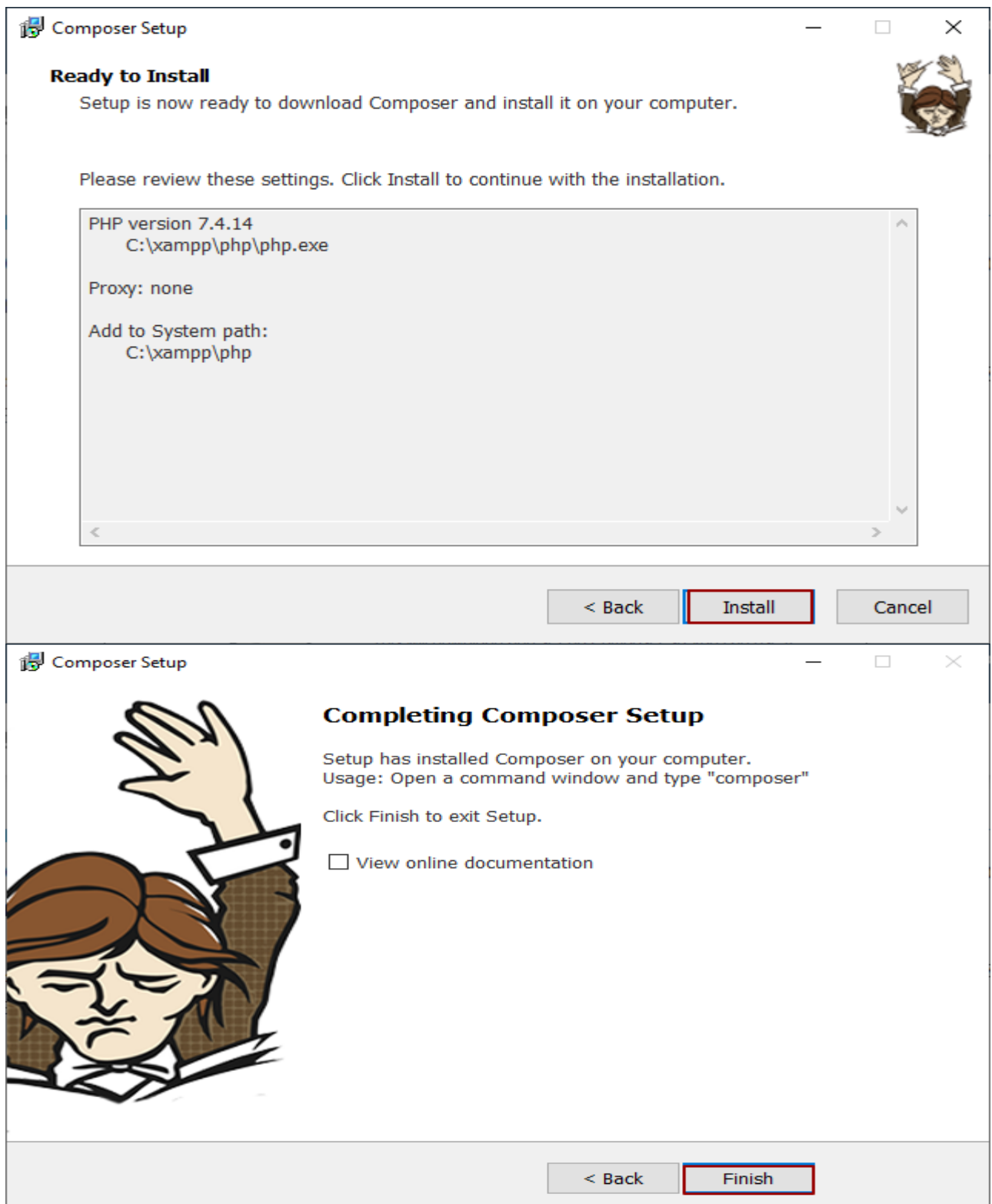
Enter proxy url

< Back

Next >

Cancel





## STEP 5: Install Magento2 Dependencies Using Composer.

Now execute the below **commands** in the command prompt to **downgrade the composer version** and install the required **Magento2 dependencies** on your local host.

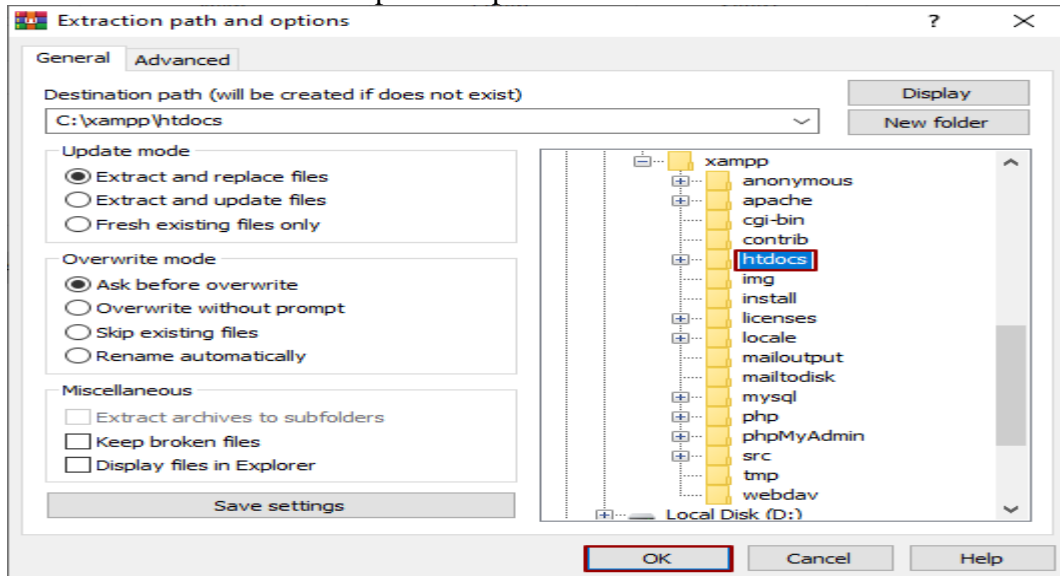
- ```
cd C:\xampp\htdocs\magento2
1 composer self-update 1.10.17
2 composer update --ignore-platform-reqs
3 composer install
4
```

## STEP 6: Similarly, Download And Install The Elastic Search To Move Further.

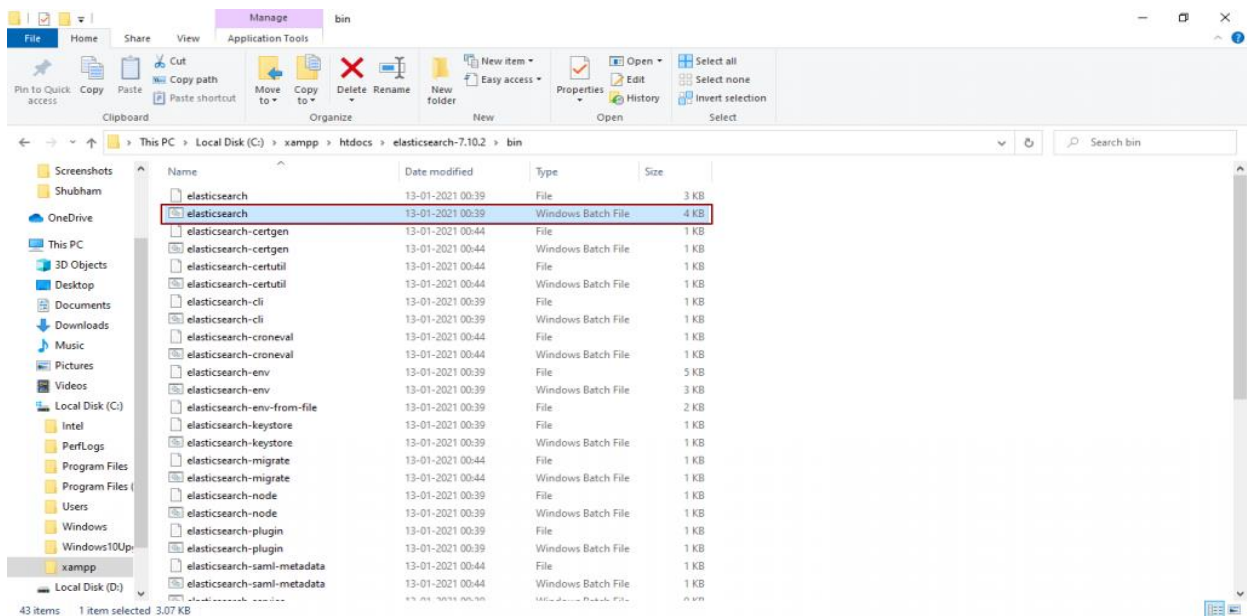
Download Elasticsearch from the mentioned

link: <https://www.elastic.co/downloads/elasticsearch> and install it.

Extract the Zip file in the htdocs folder of the XAMPP installation directory after it has been downloaded. To do that, right-click the elasticsearch zip file and select extract files, and then select the htdocs destination path and press OK.



After successful extraction, run the bash file as an administrator.



You will get the output below.

```
C:\WINDOWS\System32\cmd.exe
[2021-01-29T16:38:09,663][INFO ][o.e.e.NodeEnvironment ] [DESKTOP-SIGRVC2] using [1] data paths, mounts [[(C:)], net
usable_space [8.3gb], net total_space [97.1gb], types [NTFS]
[2021-01-29T16:38:09,663][INFO ][o.e.e.NodeEnvironment ] [DESKTOP-SIGRVC2] heap size [1gb], compressed ordinary objec
t pointers [true]
[2021-01-29T16:38:10,026][INFO ][o.e.n.Node ] [DESKTOP-SIGRVC2] node name [DESKTOP-SIGRVC2], node ID [sJv_
vXOF5J-HWrxx_BaYSQ], cluster name [elasticsearch], roles [transform, master, remote_cluster_client, data, ml, data_conten
nt, data_hot, data_warm, data_cold, ingest]
[2021-01-29T16:38:25,807][INFO ][o.e.x.m.p.l.CppLogMessageHandler] [DESKTOP-SIGRVC2] [controller/16632] [Main.cc@114] co
ntroller (64 bit): Version 7.10.2 (Build 40a3af639d4698) Copyright (c) 2021 Elasticsearch BV
[2021-01-29T16:38:27,597][INFO ][o.e.x.s.a.s.FileRolesStore] [DESKTOP-SIGRVC2] parsed [0] roles from file [C:\xampp\htdocs
cs\elasticsearch-7.10.2\config\roles.yml]
[2021-01-29T16:38:31,600][INFO ][o.e.t.NettyAllocator ] [DESKTOP-SIGRVC2] creating NettyAllocator with the following
configs: [name=unpooled, suggested_max_allocation_size=256kb, factors={es.unsafe.use_unpooled_allocator=null, g1gc_enab
led=true, g1gc_region_size=1mb, heap_size=1gb}]
[2021-01-29T16:38:31,975][INFO ][o.e.d.DiscoveryModule ] [DESKTOP-SIGRVC2] using discovery type [zen] and seed hosts
providers [settings]
[2021-01-29T16:38:40,675][WARN ][o.e.g.DanglingIndicesState] [DESKTOP-SIGRVC2] gateway.auto_import_dangling_indices is d
isabled, dangling indices will not be automatically detected or imported and must be managed manually
[2021-01-29T16:38:42,000][INFO ][o.e.n.Node ] [DESKTOP-SIGRVC2] initialized
[2021-01-29T16:38:42,000][INFO ][o.e.n.Node ] [DESKTOP-SIGRVC2] starting ...
[2021-01-29T16:38:42,733][INFO ][o.e.t.TransportService] [DESKTOP-SIGRVC2] publish_address {127.0.0.1:9300}, bound_ad
dresses {127.0.0.1:9300}, {[::1]:9300}
[2021-01-29T16:38:43,821][WARN ][o.e.b.BootstrapChecks ] [DESKTOP-SIGRVC2] the default discovery settings are unsuita
ble for production use; at least one of [discovery.seed_hosts, discovery.seed_providers, cluster.initial_master_nodes] m
ust be configured
[2021-01-29T16:38:43,894][INFO ][o.e.c.c.ClusterBootstrapService] [DESKTOP-SIGRVC2] no discovery configuration found, wi
ll perform best-effort cluster bootstrapping after [3s] unless existing master is discovered
[2021-01-29T16:38:46,904][INFO ][o.e.c.c.Coordinator ] [DESKTOP-SIGRVC2] setting initial configuration to VotingCon
figuration{sJv_vXOF5J-HWrxx_BaYSQ}
```

Note: Please don't close this window until the completion of the Magento 2.4 installation.

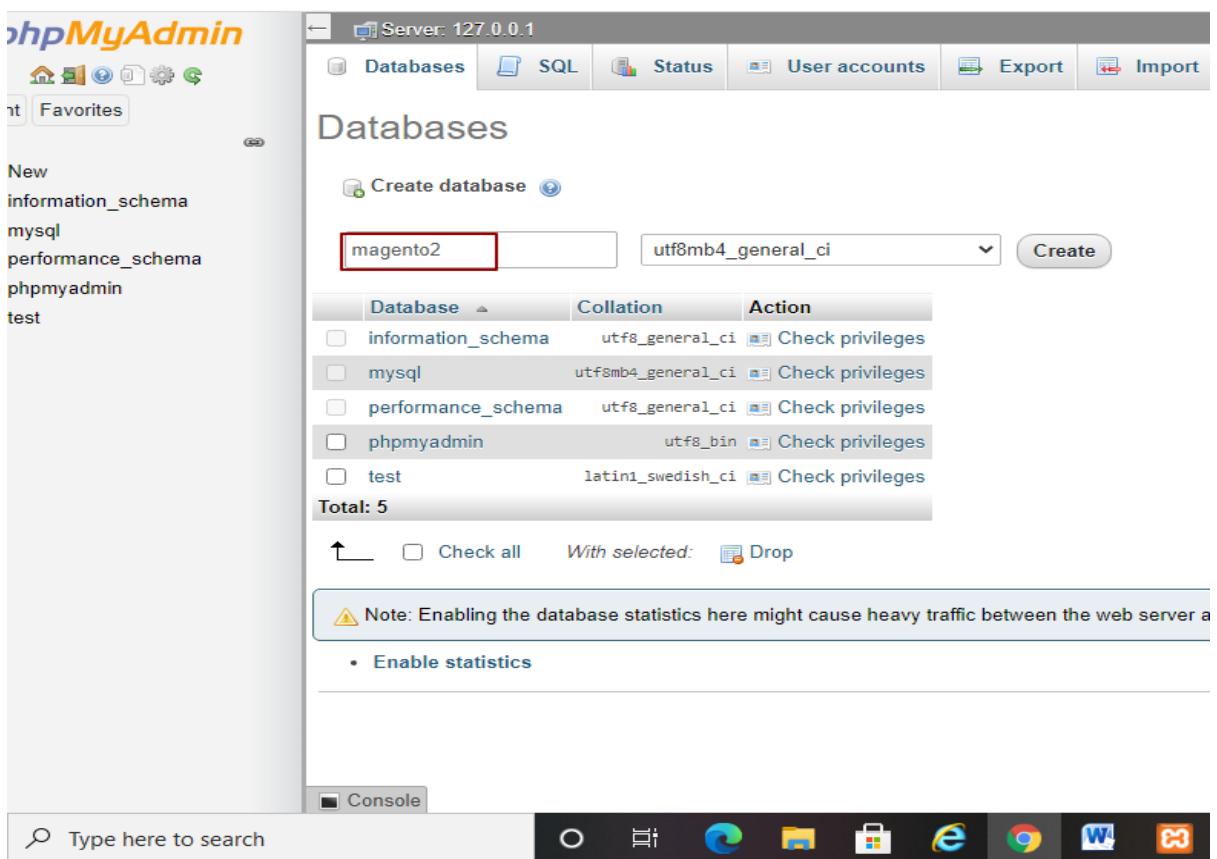
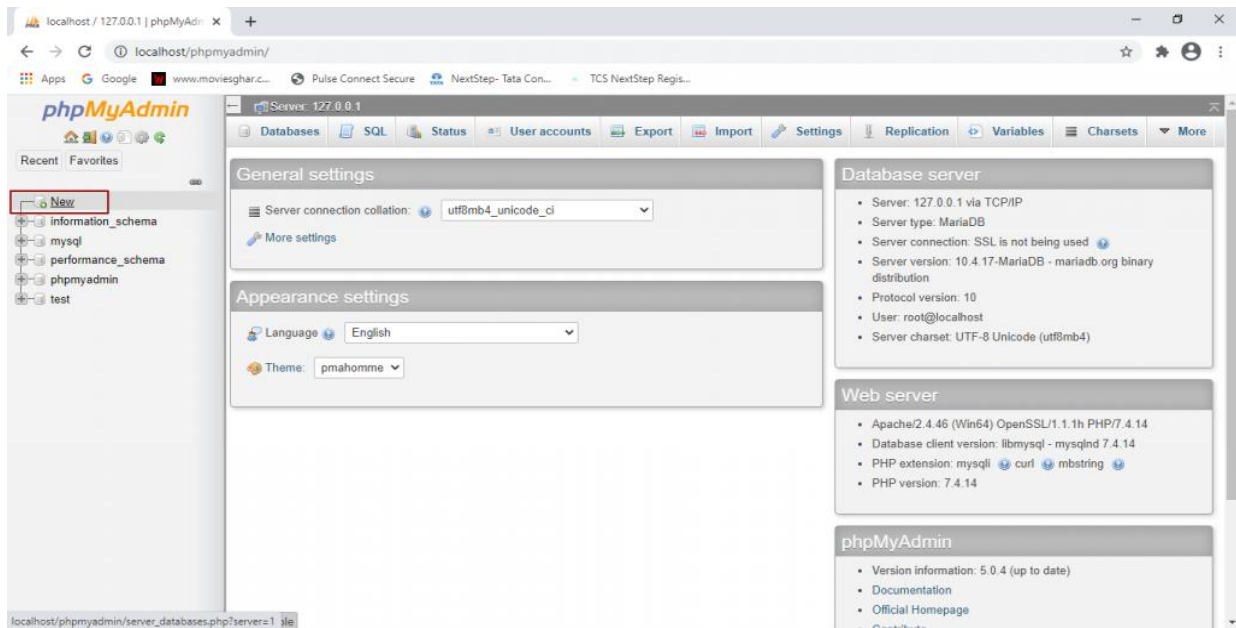
Otherwise, it will show up an error.

Please wait for a while to get the installation done. Then type "localhost:9200" in the browser and you'll get details of the installed Elasticsearch extension. The output will be identical as seen in the image below:

```
localhost:9200
{
  "name": "DESKTOP-SIGRVC2",
  "cluster_name": "elasticsearch",
  "cluster_uuid": "zRTmdYyGTP2dhzb3EchHoQ",
  "version": {
    "number": "7.10.2",
    "build_flavor": "default",
    "build_type": "zip",
    "build_hash": "747e1cc71def077253878a59143c1f785afa92b9",
    "build_date": "2021-01-13T00:42:12.435326Z",
    "build_snapshot": false,
    "lucene_version": "8.7.0",
    "minimum_wire_compatibility_version": "6.8.0",
    "minimum_index_compatibility_version": "6.0.0-beta1"
  },
  "tagline": "You Know, for Search"
}
```

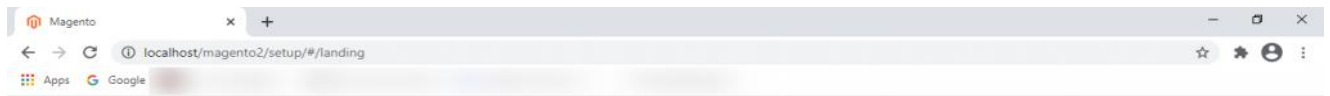
## STEP 7: MySQL Database Creation.

For Magento2, we have to create a database. Firstly, open the web browser and hit the URL "<http://localhost/phpmyadmin>" to open the PHPMyAdmin page. Then, by using the "new" button in the menu section on the left, create a database by the name "magento2" or by another name of your choice.



## STEP 8: Magento Installation.

After creating the database, the next step is to install Magento2. Open your browser first and enter "http://localhost/magento2". You will get the output as similar below.



Version 2.4.1

Welcome to Magento Admin, your online store headquarters.  
Please review [Terms & Agreement](#) and read [Getting Started](#) to learn  
how to install Magento using the command line.

Likewise, **Execute The Commands Below To Install Magento 2** By Using The Required Variables Such As Host, Database Name, Username, Password, Etc.

```
php bin/magento setup:install --base-url="http://Public IP Address or Domain Name" --db-host="localhost" --db-name="magento2" --db-user="root" --admin-firstname="admin" --admin-lastname="admin" --admin-email="user@example.com" --admin-user="admin" --admin-password="admin123" --language="en_US" --currency="USD" --timezone="America/Chicago" --use-rewrites="1" --backend-frontname="admin" --search-engine=elasticsearch7 --elasticsearch-host="localhost" --elasticsearch-port=9200
```

When this command is successfully executed, the output will be identical as seen in the image below:

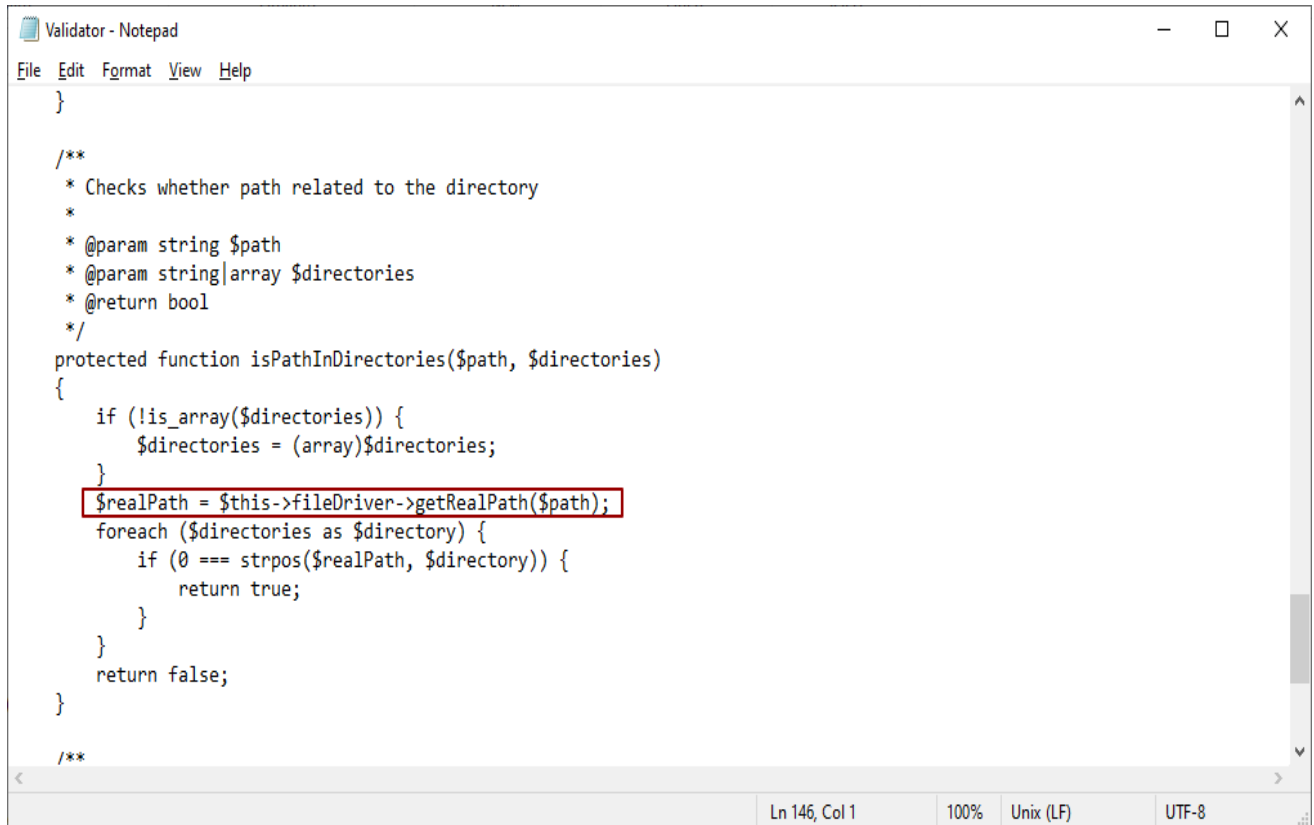
```
Command Prompt
[Progress: 802 / 818]
Module 'Magento_Version':
[Progress: 803 / 818]
Module 'Magento_Swagger':
[Progress: 804 / 818]
Module 'Magento_MagentoSync':
[Progress: 805 / 818]
Module 'Magento_MagentoSecurity':
[Progress: 806 / 818]
Module 'Magento_Mage':
[Progress: 807 / 818]
Module 'Magento_MagentoGraphQL':
[Progress: 808 / 818]
Module 'Magento_CatalogWidget':
[Progress: 809 / 818]
Module 'Magento_Wishlist':
[Progress: 810 / 818]
Module 'Magento_WishlistAnalytics':
[Progress: 811 / 818]
Module 'Magento_WishlistGraphQL':
[Progress: 812 / 818]
Enabling caches:
Current status:
Layout: 1
Block_html: 1
Full_page: 1
[Progress: 813 / 818]
Installing admin user...
[Progress: 814 / 818]
Caches clearing:
Cache cleared successfully
[Progress: 815 / 818]
Disabling Maintenance Mode:
[Progress: 816 / 818]
Post installation file permissions check...
For security, remove write permissions from these directories: 'C:/xampp/htdocs/magento2/app/etc'
[Progress: 817 / 818]
Write installation data...
[Progress: 818 / 818]
[SUCCESS]: Magento installation complete.
[SUCCESS]: Magento Admin URI: /admin
Nothing to import.
C:\xampp\htdocs\magento2>
```

## STEP 9: Now, Open The “Validator.php” File Located As Per The Following Path:

magento\lib\internal\magento\framework\View\Element\Template\File\Validator.php

### Find these lines;

---



```
Validator - Notepad
File Edit Format View Help
}

/**
 * Checks whether path related to the directory
 *
 * @param string $path
 * @param string|array $directories
 * @return bool
 */
protected function isPathInDirectories($path, $directories)
{
    if (!is_array($directories)) {
        $directories = (array)$directories;
    }
    $realPath = $this->fileDriver->getRealPath($path);
    foreach ($directories as $directory) {
        if (0 === strpos($realPath, $directory)) {
            return true;
        }
    }
    return false;
}

/**
```

and then modify it to,

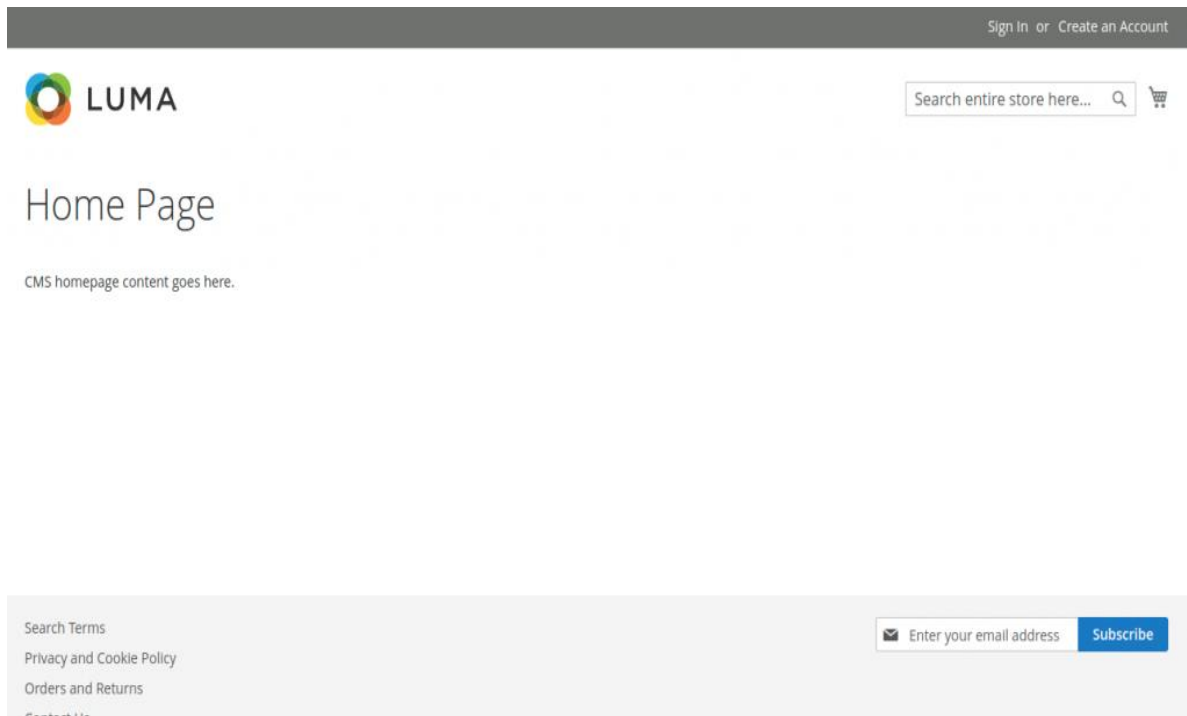
```
$realPath = str_replace('\', '/', $this->fileDriver->getRealPath($path));
```

and at last, **save the file** and exit.

## STEP 10: In Addition, Enable Requisite Commands.

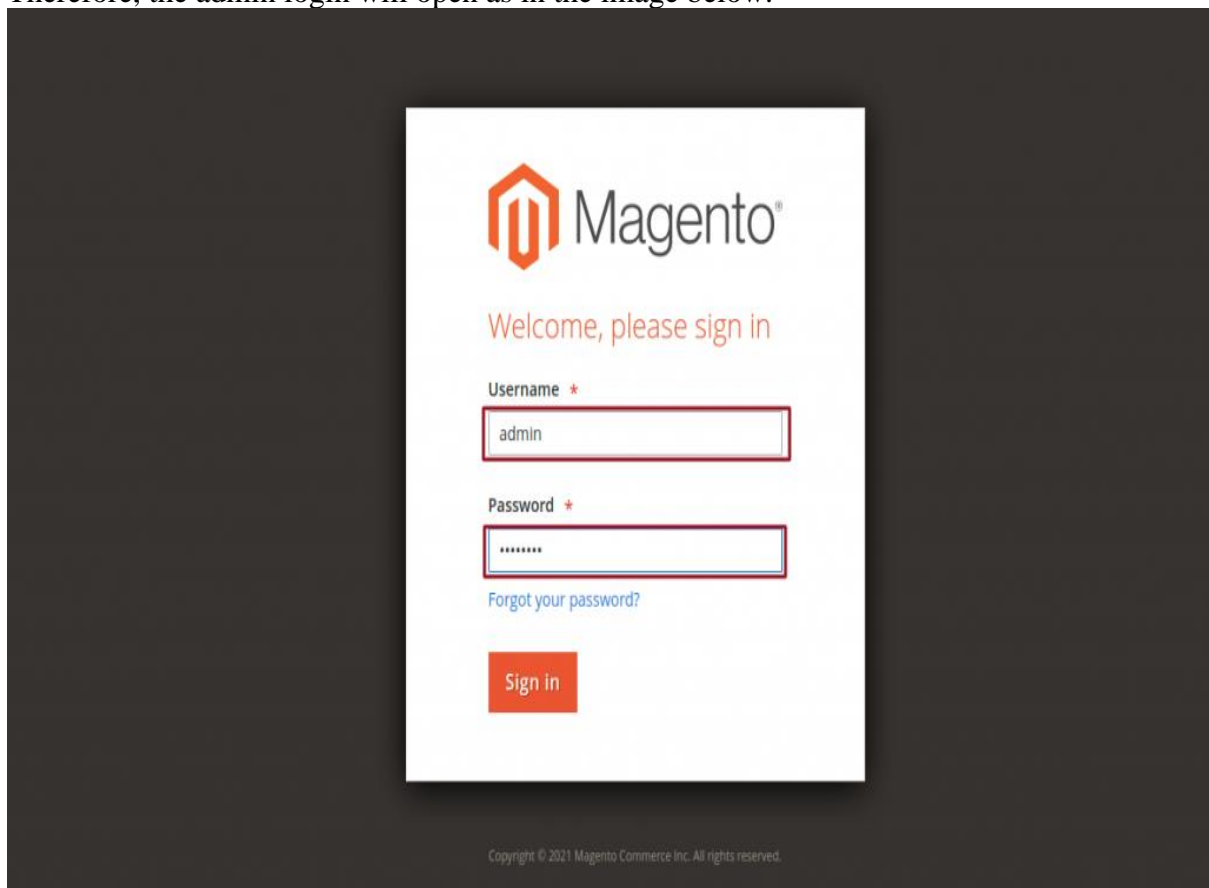
- 1 php bin/magento indexer:reindex
- 2 php bin/magento setup:upgrade
- 3 php bin/magento setup:static-content:deploy -f
- 4 php bin/magento cache:flush

While the installation of Magento2 has been completed. Go to your **browser** and enter the **IP address or your domain name**. As a result, the homepage of your **default E-Commerce website** will be like this:



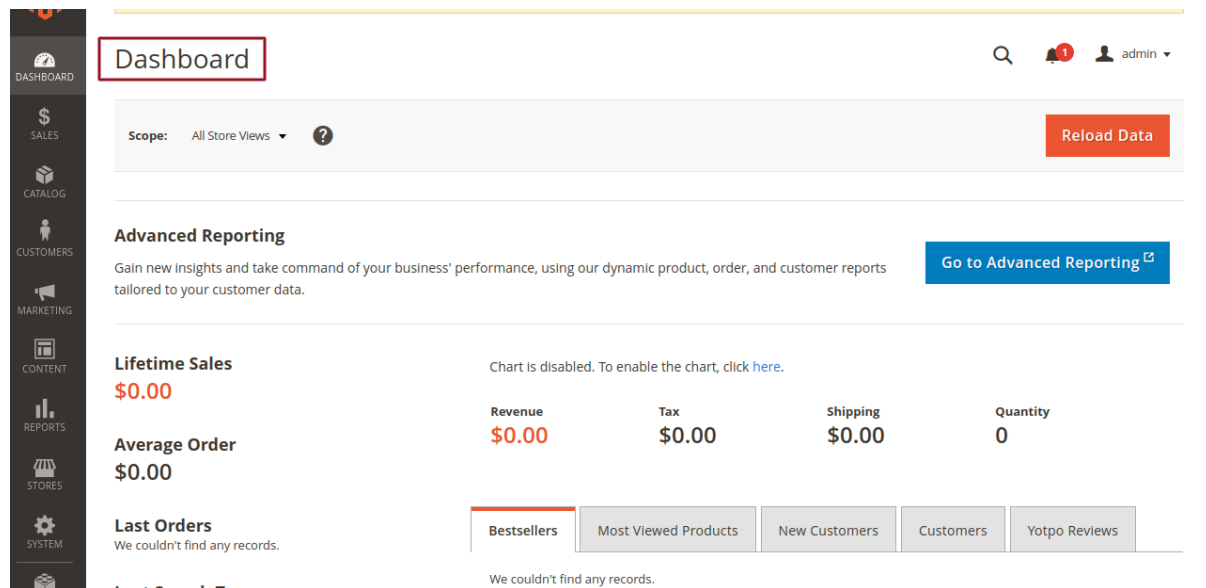
### STEP 11: Magento Configuration.

After all the above steps, you can now access your admin panel by using your credentials. Therefore, the admin login will open as in the image below.





After installation completed, default admin dashboard looks like.



Magento E-Commerce platform is ready to use.

## Need of composer

A composer is basically a tool for dependency management for PHP language and in Magento 2, Composer plays a key role in gathering components and product editions. It basically allows developers to declare the libraries that your project depends on and it installs, updates, and manages them for user. Magento 2 uses a composer for dependency management. It is used to manage the Magento components and their dependency. Composer allows us to declare the libraries on which our project depends, and it will also manage install/update for us.

**For Example** - In our daily life, we create projects and then use the third party libraries to manage the dependencies as for sending Email or for creating PDF. We create our projects at these libraries, and a whole bunch of code becomes very wide. But in that code, we have a minimal part which has been written by us and the other part is the third party library.

Let's suppose, we want to shift this code from one system to another, that time we are not shifting only our code but also dependency that we have used in that. These dependencies are widely available, we should not ship them from one place to another, and only our code should be shifted, and then the dependency should automatically get on to another system. So, to manage this, PHP's composer is very widely used.

There are two important commands of Composer, which are given below:

**Composer install** - It will be run at the very first time when we install any project. By using this command, all the dependencies are fetched.

**Composer update** - It will check that what has been updated based on the composer. Lock file. The composer. lock file contains the information that what was installed previously.



## Composer necessary

We can install the Magento 2 without having a composer, but it should be installed using composer. Composer reads composer.json file in the root directory of Magento to download the third-party dependencies listed in the file. Installation of Magento 2 is important using composer because whenever we install Magento using composer, it will create a file, i.e., composer.json. This file gives the information that what Magento version we are using, and a couple of other dependency details. These dependencies are automatically loaded based on the Magento version. So, if there is any update release, then we just only need to update the Magento version in composer.json file. Magento will accept the dependencies according to the Magento Community Edition. So, we do not need to worry about the upgrades.

Composer is very important because all the upgrades that we will do in Magento will be done only using composer. For Magento 2 installation using Composer.

## Advantages of using composer

Installation of Magento 2 using composer offers the following advantages:

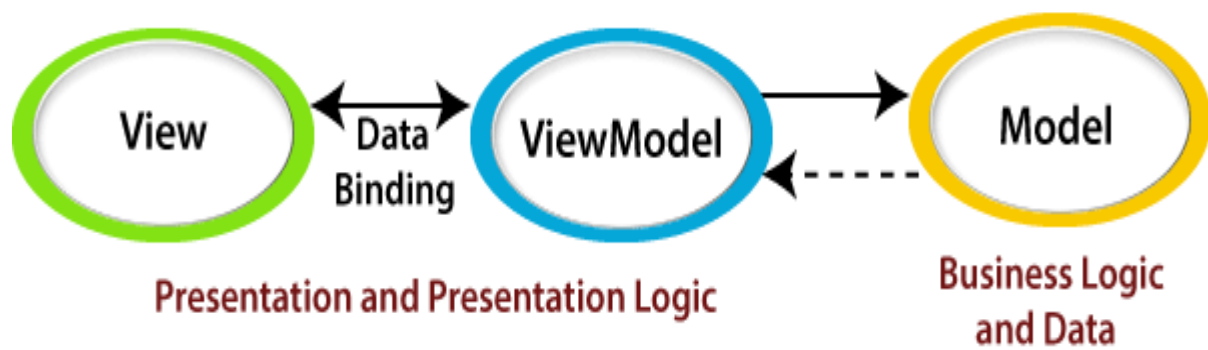
- Without bundling the third-party library with source code, we can reuse them.
- Reduces the extension conflicts and compatibility issues by using a component-based architecture with robust dependency management.
- Repackage the Magento Open Source with the other components.
- Follow the PHP Framework Interoperability Group (FIG) standards.
- Use the Magento software in a production environment.

**Autoloading** is an important feature of the composer. Based on the autoload section in composer.json file, composer generates an **autoload.php** file inside the vendor directory. It ties into the bootstrap process you can see in index.php under the Magento 2. In short, it automatically includes the required code throughout your store.

## Magento Architecture and Frame Work

Magento 2 has a totally different architecture than Magento 1. Its architecture is designed with the objective of making the source code as an extensive and modularized as possible. The main purpose of this approach is to allow it to be easily adapted and customized according to the need of the project. Every business logic and functionality is designed in the form of modules. So, every module can work as a standalone unit.

Magento 2 has a **Model View ViewModel (MVVM) architecture**. This MVVM architecture provides much more robust separation between the Model and View layer, as it is closely related to the Model View Controller (MVC). The brief description of MVVC is given below:



### Magento Architecture

**Model:** It holds the business logic of the application, and depends on an associated class - Resource Model for database access. Similar to the MVC, it is the logic of data management and description of the fundamental data which is necessary for the operation of the application. The model responds to the request coming from the view.

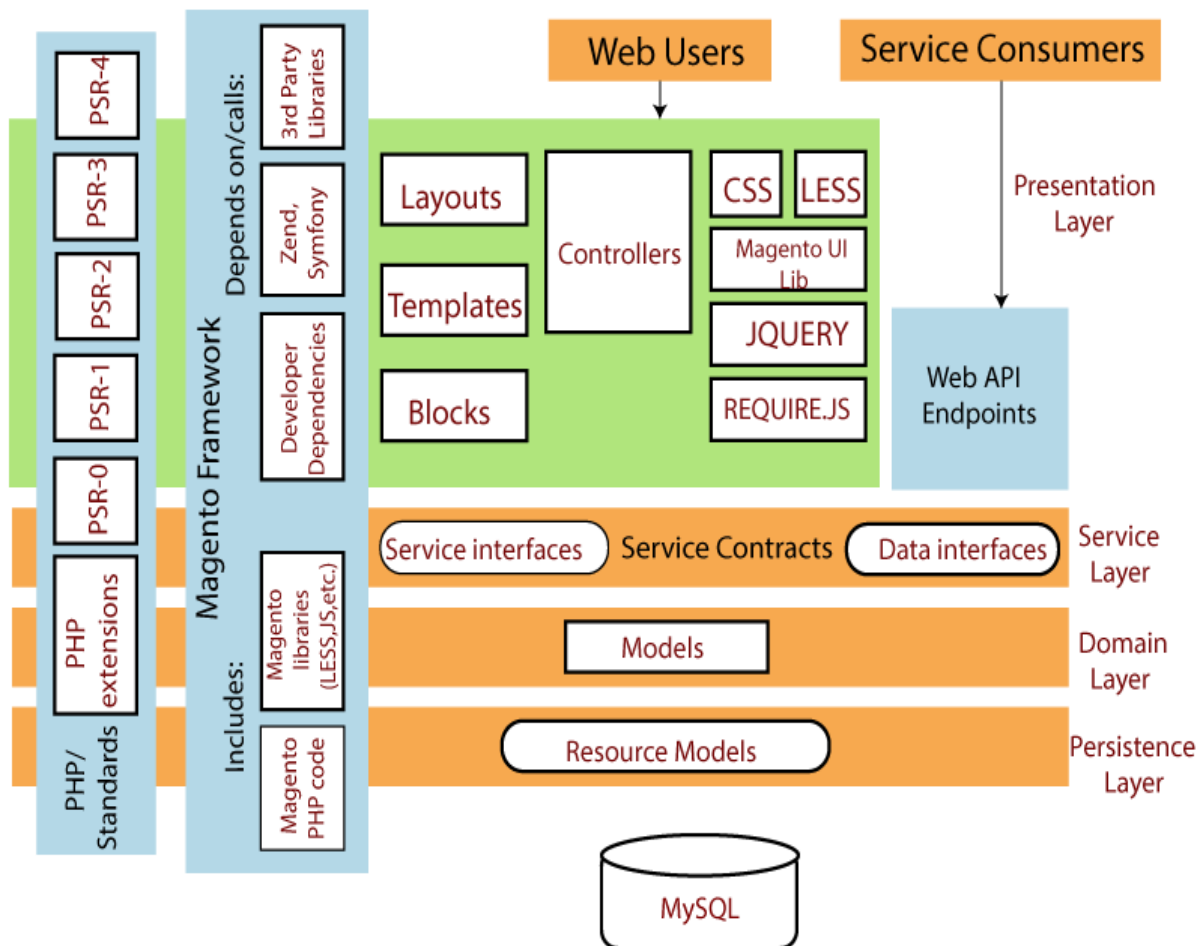
**View:** The view is a graphical interface that the users see on the screen. It is responsible for displaying the respond for the user request. The view is a structure or a layout, which represents the data in a particular format. Views specify that "how your data looks like". It is a subscriber for the changes of the property values or commands provided by the ViewModel.

**ViewModel:** The ViewModel interacts with the Model layer and exposes only necessary information to the View layer. This is handled by module's Block classes in Magento 2. An important thing to notice that this was usually part of the Controller role of an MVC system. In MVVM, the Controller is only responsible for handling the user flow means it receives requests and either tell the system to render a view or to redirect the user to another route. It contains a Model which is converted to a View and also contains the commands that can be used to affect the Model. The ViewModel is basically an abstraction of the view exposing public properties and commands.

### Magento Frame Work

Magento 2 software is a CMS (Content Management System), developed by Varien Inc. It is an open-source software which is a very useful software for online business. Magento 2 is developed in PHP and Zend framework. Magento 2 is currently the largest E-commerce platform in the world. Magento 2 is known for easy customization and extension of its functionalities.

Magento 2 is split into four layers, according to the official documentation.



## Magento Frame Work

The detailed description of Magento 2's layer architecture is given below.

### 1) Presentation Layer

The presentation layer is the uppermost layer of Magento 2 architecture. When we interact with the web interface of Magento, that time, we are interacting with the presentation layer code. It contains all controllers and View elements such as - layouts, templates, block, and css.js, etc. The presentation layer calls the service layer using service contracts, usually. It can overlap business logic.

Web users, system administrators, are the user of the presentation layer

## 2) Service Layer

The service layer is the middle layer among the presentation layer and domain layer. It provides a bridge between the presentation and domain layer and resource-specific data. The service layer implements service contracts, which are defined using the PHP interface. These service contracts allow us to add or change the business logic resource model without breaking the system. This is done using the dependency injection configuration file (di.xml).

The service layer grants access to API (REST/SOAP or other modules). It resides above the domain layer and below the presentation layer.

## 3) Domain Layer

The domain layer is responsible for business logic. It does not contain database-specific or resource-specific information. The domain layer defines the generic Magento data objects or models which contain business logic. This logic describes that which operation can be performed on the particular types of data, such as Customer object.

To retrieve the data from the database with the MySQL calls, each domain-layer model has a reference to a resource model.

## 4) Persistence Layer

It describes the resource model, which is responsible for data extraction and modification in the database using the CRUD (Create, Read, Update, and DELETE) requests. For completing these requests, the resource model contains the SQL code. Additional business logic capabilities are also implemented.

**For example** - database functions implementation and data validation.

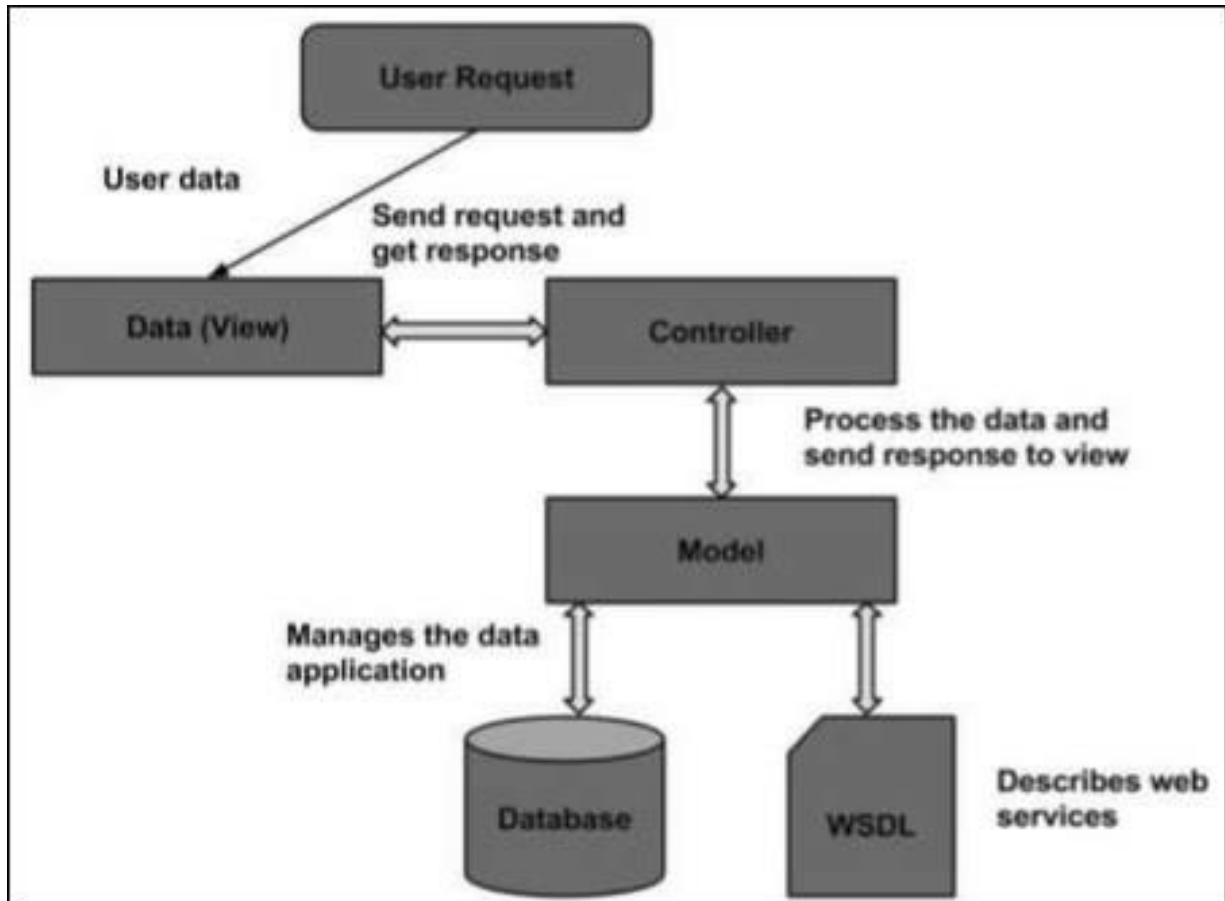
As a software developer, it is crucial to write clean, maintainable, and compatible code. Adhering to the PHP Standard Recommendation (PSR) guidelines will ensure your PHP code is consistent and easy to read. PSR standards are coding guidelines created by the PHP Framework Interoperability Group (PHP-FIG). The main goal of these standards is to streamline the development process by defining a set of rules for how PHP code should be written, formatted, and organized. By following these rules, PHP developers can ensure that their code is more readable, maintainable, and compatible with other PHP codebases.

**Example:** PSR-0 – Autoloader Standard, PSR-1 – Basic Coding Standard, PSR-2 – Coding Style Guide, PSR-3 – Logger Interface, PSR-4- Autoloader Standard

## Magento Block Diagram

Magento block diagram provides a high-level overview of Magento's architecture, showing how different components interact to deliver a robust e-commerce solution. It's important to note that the actual architecture may vary depending on the specific implementation and customization of the platform.

Architectural style of Magneto, for implementing user interfaces. The following block diagram shows the magneto functionality The Magneto architecture comes with Models, Views and Controllers.



**Block diagram**

**User Request** – The user sends a request to a server in the form of request message where web browsers, search engines, etc. act like clients.

**View** – View represents the data in particular format. It is the user interface which is responsible for displaying the response for user request. It specifies an idea behind the presentation of the model's data to the user. Views are used to reflect "how your data should look like".

**Controller** – The controller is responsible for responding to user input and perform interactions on the data model objects. It uses models to process the data and send responses back to the view.

**Model** – The model is responsible for managing the data of the application. It contains logic of the data and represents basic data object in the framework. It responds to request from the view and to the instructions from the controller to update itself.

**Database** – Database contains the information which is requested from the user. When the user requests data, view sends requests to the controller, the controller requests from the model and the model fetches the required information from the database and responds to the user.

**WSDL** – WSDL stands for Web Services Description Language. It is used for describing web services and how to access them.

## Magento Folder Structure

A folder is the virtual location for applications, documents, data or other sub-folders. Folders help in storing and organizing files and data in the computer. Collections of files are stored in directories. These collections of files are often related to each other; storing them in a structure of directories keeps them organized

### Magento 2 Folder structure

If you navigate to your Magento 2 root, you'll see the next folders:

- `app`;
- `bin`;
- `dev`;
- `lib`;
- `pub`;
- `var`;
- `vendor`.

Each folder consists of subfolders where the subfolders contain files. Let's have a closer look at each of the folders:

#### `/app`

As well as in Magento 1, the folder contains the main Magento code;

- it includes the `/app/design` with subfolders;

`adminhtml` and `frontend` directories;

the `/app/design/adminhtml` and `/app/design/frontend` directories contain theme folders with various **CSS** and **LESS** files;

- the `/etc` folder contains the **di.xml** configuration file;

the **di.xml** file contains class mapping and interface preferences;

- the `/app` directory may contain an `/app/code` subfolder (This depends on your install method. The code directory will be there in case you download the GitHub repository. Otherwise, all Magento 2 code will be contained in the vendor folder.);
- if you have the `/app/code` subfolder within `/app`, it includes the Magento core code (One of the locations where you can develop your custom Magento code. The info is actual in case you don't use composer to handle your custom code.).

#### `/bin`

- the directory contains the **magento** file or a **CLI** script;
- the **CLI** script activates a console with a set of handy options: enable/disable cache; enable/disable modules; run an indexer, etc.

## **/dev**

- Magento 2 test scripts and a few tools for development are placed in the folder.

## **/lib**

- the directory contains several libraries packed separately from the Magento core;
- the **internal** and **web** subfolders are also located here;
- the **internal** directory contains **server-side libraries** and several **font** files;
- the **web** subfolder contains **client-side** (JavaScript) **libraries**, **jquery**, **modernizr**, **requirejs**, **prototype**, **scriptaculous**. (The framework code will be included in the lib folder only in case you download the GitHub repository. Otherwise, Magento 2 code will be in the vendor folder.).

## **/pub**

- it should be set as your web root on the server you use (Apache, Nginx);
- as you have noticed from the name the folder contains all the publicly accessible files;
- **index.php** controller is also contained in the folder.

## **/var**

- caches and generated classes are all located in the folder;
- the **var** folder includes **cache**, **composer\_home**, **generation**, **log**, **di**, **view\_preprocessed**, **page\_cache** files.

## **/vendor**

- the folder is generated by composer using **composer.json** file;
- the folder includes various packages that have been defined under **composer.json**.

*Note that in Magento 2.2.x and later versions the folder **var/generation** was removed to the project root and renamed to **generated**.*

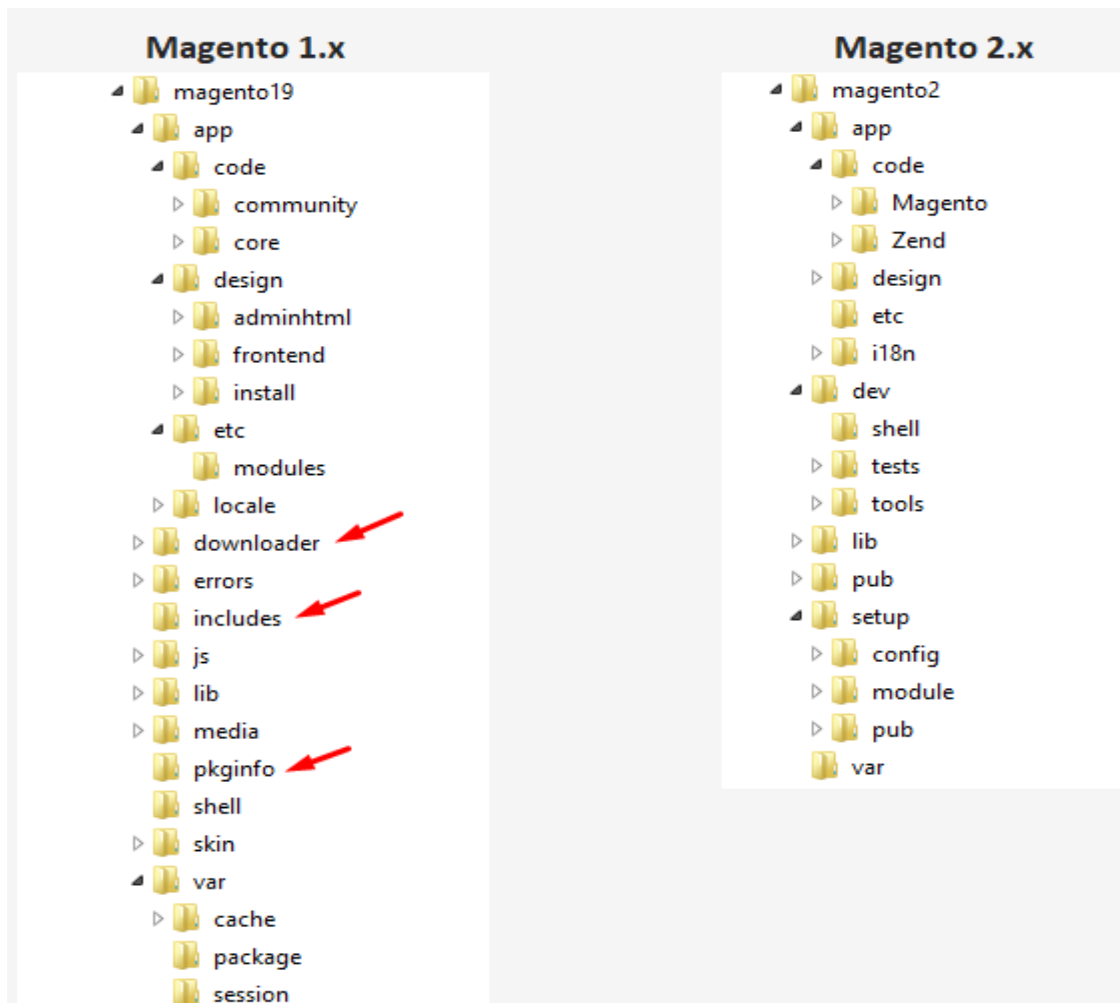
## **Magento 1/ Magento 2 difference in Folder structure**

The **app**, **lib** and **var** directories were included in Magento 1 too. Magento 2 contains such new directories as **bin**, **dev**, **phpserver**, **pub**, **setup**, **update**, and **vendor**. For those who have just migrated from the first version, here is a table with the basic purpose of the new directories:

**Table -2 Magento Directory and purpose**

| <b>directory</b> | <b>purpose</b>                                                                 |
|------------------|--------------------------------------------------------------------------------|
| <b>App</b>       | includes the design folder for themes; the configuration file <b>env.php</b> ; |
| <b>Bin</b>       | includes the Magento 2 command line tool named magento;                        |
| <b>Dev</b>       | includes testing files for the Magento 2 Functional Testing Framework;         |
| <b>Lib</b>       | includes various libraries and packages ;                                      |
| <b>phpserver</b> | includes <b>router.php</b> ;                                                   |
| <b>Pub</b>       | includes generated static files, the website media;                            |
| <b>Setup</b>     | the performance toolkit and other important packages;                          |
| <b>Update</b>    | similar to the downloader from Magento 1 and contains upgrade files;           |
| <b>Var</b>       | similar to Magento 1 and includes raw cache files,logs, and reports;           |
| <b>vendor</b>    | the closest to app/etc/modules.                                                |





- now **media**, **error**, **skin**, **js**, **get.php**, **cron.php**, and **index.php** folders are placed to **pub** folder;
- **skin** and **js** folders are put in **pub/static** folder. Static content can be maintained easier via CDN now.;
- **js**, **css**, **images** are divided into **pub/static/adminhtml** and **pub/static/frontend** (for backend and frontend correspondingly);
- if you are looking for migration **tools** and **tests**, as well as **shell**, have a dip into the **dev** folder;
- **setup** directory is now used for installation process;
- **downloader**, **includes**, **pkginfo** have been excluded from Magento 2.

## Magento Design Concept and Terminology with scenarios

Magento is a powerful E- Commerce platform that allows for the highly customized and personalized shopping experiences that customers expect when shopping online.

### WEBSITES AND STORES DESIGN

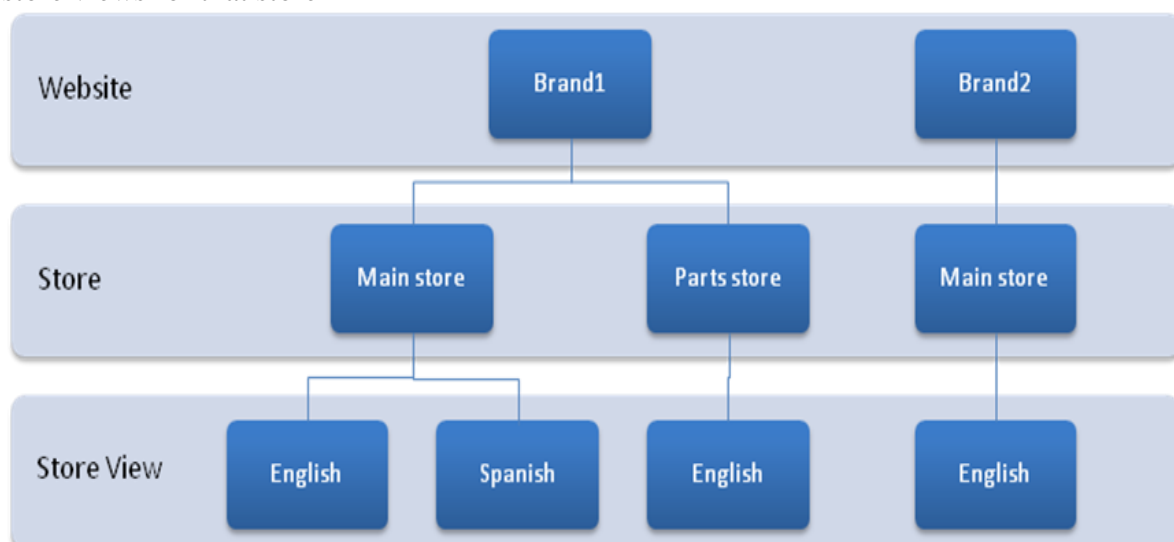
Magento natively supports the creation and management of multiple stores in a single Magento installation, Magento has a hierarchy of concepts that define the relationship between the individual stores in a Magento installation.

In Magento, a website is a collection of stores, which themselves are collections of store views. These layers, although perhaps initially confusing, provide you with powerful flexibility when setting up online businesses in Magento.

**Website:** A **website** is made up of one or more stores which share the same customer information, order information and shopping cart.

**Store:** **Stores** are collections of store views and can be setup in a variety of ways. Their main function is to provide a logical container that allows you to group related store views together in a website.

**Store View:** **Store Views** are the actual store instances in Magento. Most stores will have a single store view associated with them. But a store can also have multiple store views, which are typically used for different languages. Therefore, if you wanted to have a store displayed in English and Spanish, for example, you could create the store once and create two different store views for that store



Hierarchy of websites, stores and store views in Magento.

## Terminology with scenarios

### SCENARIO 1- ONE STORE

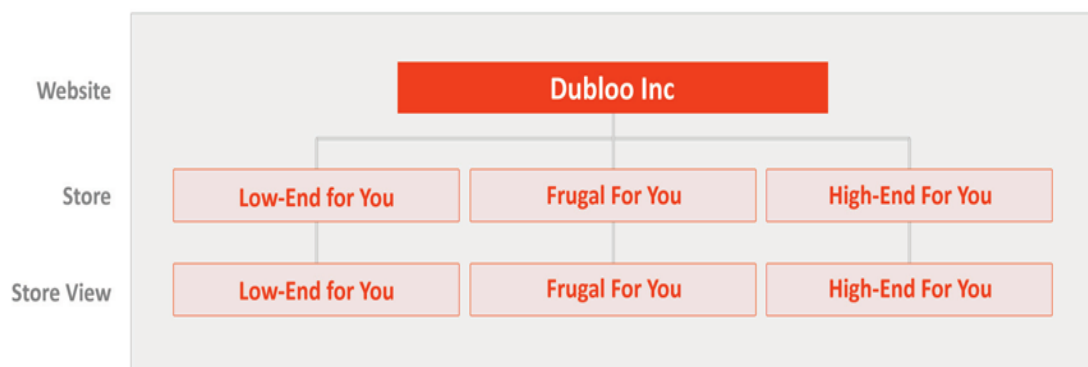
A company named “Bongo’s Instruments” wants to create an online presence. Bongo has a single catalog and does not need to support multiple languages. This is the simplest scenario in which Bongo’s Instruments is the website, store, and store view.



Single website, store and store view

### SCENARIO 2 - MULTIPLE RELATED STORES

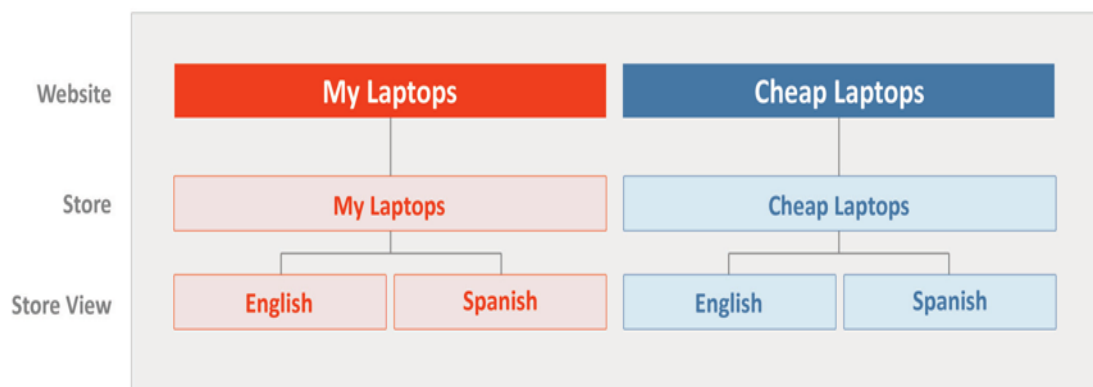
A company named “Dubloo” creates an online presence with three separate clothing stores that each cater to a different price-level audience. Dubloo wants the ability for all three of its stores to share customer and order information - meaning customers can create an account in any of the stores and it will be available to them in the others as well. In this scenario, Dubloo would have one website and three stores under their online presence. Because all of the stores support a single language, they each have only one store view.



Single website with multiple stores.

### SCENARIO 3—MULTIPLE INDEPENDENT STORES

A company named “My Laptops” wants an online presence with two separate stores that both sell laptops but at different prices and with different product selections in some categories. They also want to offer English and Spanish language options per store. Within each store they need to synchronize customer and order information, but they do not need to share this information between the two stores. In this scenario, My Laptops would have two websites (which stops customer and order info from being shared with stores in the other website), each with one store and two store views (one for English and one for Spanish).

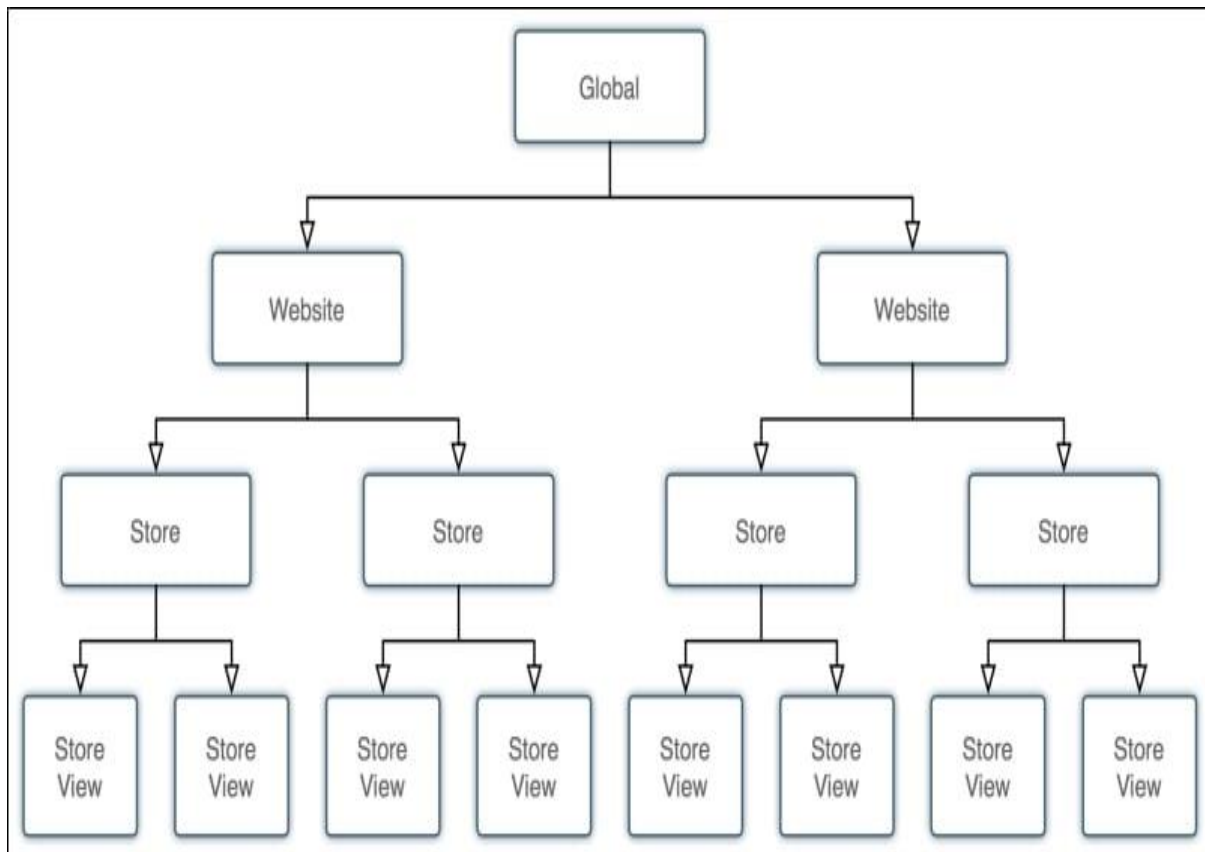


Multiple websites with multiple stores and store views

### Global – Website - Store Methodology

Magento Commerce and Magento Open-source have four levels of hierarchy: Global, Website, Store, and Store View. It is crucial to have the basic fundamental knowledge about website, store, and store-view for starting an online eCommerce store. Before you begin laying out your master plan for the various websites and stores you intend to create (and even if you're only beginning with one website), you need to master the Magento methodology for multiple stores. Magento describes this as "GWS," which stands for "Global, Website, Store." Each Magento installation automatically includes one of each part of this hierarchy, plus one more for "Store View."

The following diagram shows how each part of GWS is related to one another:



## Global - Website – Store in Magento

### Global

**Global** refers to settings (for example, stock management rules) and values (for example, product price) for the entire installation. Throughout your Magento installation, you'll find **Global** displayed next to various form fields.

In terms of installation planning, your *Global* considerations should include:

1. Will customers be shared among all sites? You can elect not to give customers the ability to register for one website and automatically be registered to all others.
2. Can I allow any user with Admin permissions to see all orders and customers from all websites and stores within the single installation? Without modification, Magento does not allow you to set up Admin users by limiting them to certain websites and stores. If an Admin user can see orders, they can see all orders for all customers.
3. Will all stores within an installation use the same rules for managing inventory? Inventory rules, such as whether stock is to be managed or whether backorders are allowed, are system-wide choices. (These choices can be changed, in some cases, at the product level, though that does mean paying careful attention to how products are configured and managed.)

In general, we recommend that you consider a single Magento installation only for multiple websites and stores that are similar in concept. For example, if your online business is selling drop-shipped furniture through several differently branded websites, then a single Magento installation is ideal. However, if you have two or more different businesses, each with a different product focus, company name, banking, and so on, it is best to use a separate Magento installation for each discrete business.

## Website

The website is the "root" of a Magento store. From the website, multiple stores are created that can each represent different products and focus. However, it is at the website level that certain configurations are applied that control common functions among its children stores and Store Views.

As described above, one of the most important considerations at the website level is whether or not customer data can be shared among websites. The decision to share this information is a Global configuration; however, remember that you cannot elect to share customer data among some websites and not others: it's an all or nothing configuration.

### Tip

If you do need to create a group of websites among which customer data is to be shared, and create other websites among which the data is not to be shared, you will need more than one installation of Magento.

## Store

What can sometimes be confusing is that "**Store**" for Magento is used to describe both a store structure as well as a Store View. When configuring your hierarchal structure, "Store" is used to associate different product catalogs to different stores under a single "Website," whereas "Store Views" can be created to display a "Store" in multiple languages or styles, each with their own URL or path. Each Store View can be assigned different themes, content, logos, and so on.

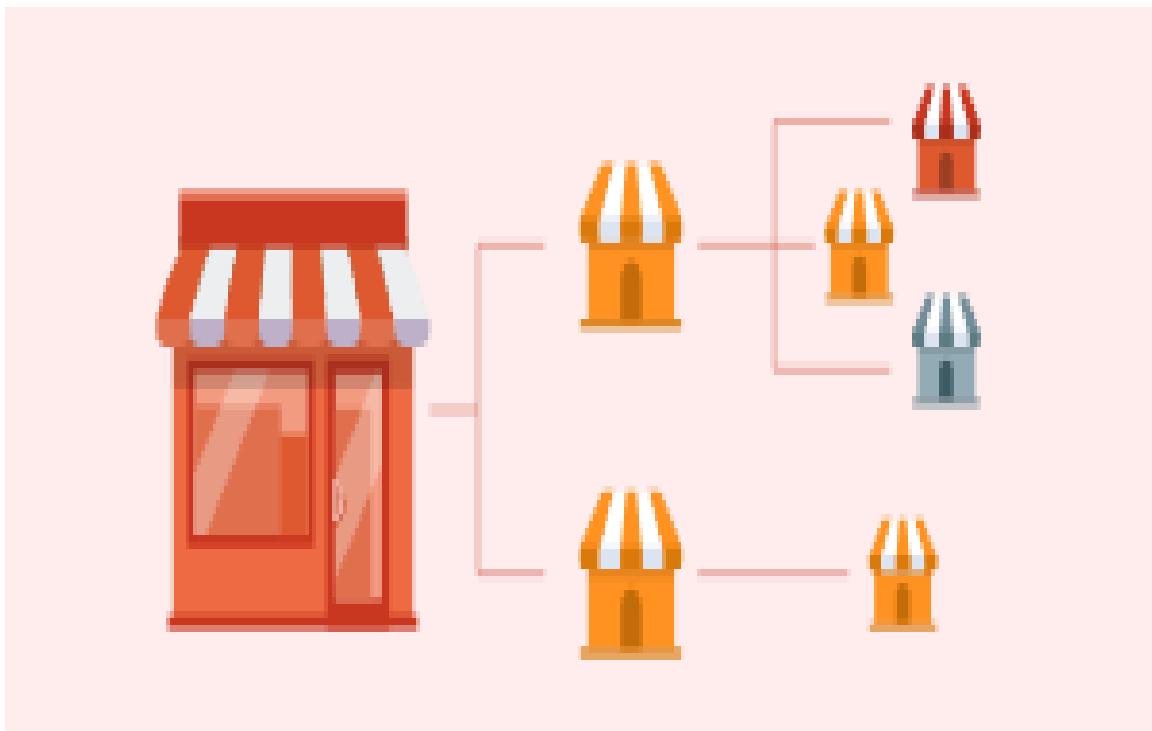
Yet, throughout Magento's many administration screens, you will see that "Store" is used to define the scope of a particular value or setting. In these instances, entered values will affect all Views under a Store hierarchy. We know this can be confusing; it was to us, too. However, by following the processes in this book, you'll quickly come to not only understand how a Store and Store View is referred within Magento, but also appreciate the tremendous flexibility this gives you.

## Tip

Perhaps the best way to consider Stores and Store Views is to learn that a View is what your website visitor will see in terms of language, content and graphics, while Store refers to the data presented in each view.

## Planning for multiple stores

Magento Multi Store is a powerful solution that allows businesses to manage multiple online stores from a single platform. It lets you create stores that share a single domain. The different stores stay under one domain and act as its subdomains. This post will highlight the challenges of Magento multi stores and practical tips to overcome them. It will also take you through the key benefits of having multiple Magento 2 stores.



With Magento 2, you can own multiple stores from a single Magento installation. The stores can be subdomains of a main website and cater to different areas of consumer demand.

Use case example: Imagine you own an online fashion retail business. You offer clothing, accessories, and footwear. With Magento Multi, you can set up three separate stores within a single Magento installation:

1. **Clothing Store:** For targeting fashion-conscious shoppers.
2. **Accessories Store:** To cater to the needs of accessory enthusiasts.
3. **Footwear Store:** For showcasing footwear that matches the clothing style you sell.

While these three stores have their own identities, they are managed centrally from one Magento admin panel. You can streamline your operations by sharing customer data, inventory, and promotions. It also provides a seamless shopping experience for your customers across all stores.

## Magento Multisite vs. Magento Multistore

| Feature          | Magento Multisite                                                                                          | Magento Multistore                                                                                                      |
|------------------|------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Creation         | Allows you to create multiple websites using a single Magento installation.                                | It allows you to manage multiple stores that sell different products on a single website.                               |
| Domain and Theme | Each website can have its unique domain name, theme, and customer base.                                    | Multiple stores can share the same domain but have different product catalogs and themes.                               |
| Customer Data    | Websites can share customer data and shopping cart information but are independent.                        | All stores on a common website share customer data and shopping cart information.                                       |
| Use Case         | Ideal for companies that operate in different regions or markets and want to keep their websites separate. | Ideal for companies that want to sell a wide range of products under different store views but within the same website. |

**Table – 3 Multisite vs. Magento Multistore in Magento**



## Benefits of creating multiple websites and stores with a single configuration

### 1. Enables Store-Specific Configurations

Magento store owners can configure different settings for their Magento 2 multiple stores. It means you can use different **payment methods** for various stores. For example, **Braintree** for one store and choose PayPal for another.

### 2. Allows Customer Segmentation

You can effectively target specific customer groups with tailored content, products, and pricing on a Magento website or store. You can create different websites and stores and target customers from various demographics.

### 3. Improves your Brand's Profitability

A multi-store setup helps reduce the costs incurred by Magento extensions and themes. It helps manage operational costs while increasing its reach and profitability.

### 4. Easy Customer Management

Customer account management becomes easy since you can ensure different store configurations. You can offer distinct payment and shipping options for different regions or create store views in **multiple languages**. It also simplifies store setup for one website.

### 5. Helps Expand your Business

With the multiple store functionality, you can create website subdomains for a main website. You can expand your business into closely related vertices. Example: A company that sells power tools can also venture into selling safety gear and utility clothing.

### 6. Allows Seamless SEO Management

You can manage the SEO settings of many stores from a common Magento 2 Admin panel. It makes monitoring and optimizing your SEO efforts easier.

### 7. Facilitates Informed Decision-Making

Utilizing a single platform streamlines data access for crucial business choices. You can monitor sales and oversee products. Also, you can seamlessly strategize inventory for all sites and stores within the admin panel.

## 8. Building Credibility

Managing multiple websites can help establish credibility and authority across various niches or sectors. It is particularly useful when consistently producing valuable and informative content on each site.

## 9. Improved User Experience

Magento multi-store allows for a user-friendly browsing experience. It offers the flexibility to design a sleek and organized site, enhancing the overall customer experience.

# Planning and Solutions of Magneto Multiple stores

### 1. Improving Performance and Scalability

When dealing with high traffic or extensive product catalogs across multiple stores in Magento, it can lead to performance challenges.

**Solution:** Implement efficient caching mechanisms, optimize your server infrastructure, and think about using **Content Delivery Networks (CDNs)** to distribute content globally. Regularly monitor your server's performance and be ready to scale up resources as needed.

### 2. Efficient Data and Configuration Management

Managing numerous stores with varying **product catalogs**, pricing, and configurations can quickly become complex and time-consuming.

**Solution:** Simplify your data and configuration management using the Shared Catalog feature. It will allow you to share common products and designs among multiple stores. Also, use CSV import/export and data migration tools to streamline the process of updating product information across your stores.

### 3. SEO and Avoiding Duplicate Content

Similar content across websites or stores can result in duplicity issues that can harm your **SEO rankings**.

**Solution:** Implement canonical tags to guide search engines toward the preferred version of a page. Ensure your robots.txt file and meta tags are correctly configured for each store view.

### 4. Managing Payment and Shipping Methods

Different stores may require unique **payment gateways** and shipping methods, complicating the management of multiple configurations.

**Solution:** Simplify the process using extensions that allow you to set specific payment and shipping methods for each store or Store View. Magento offers a range of third-party extensions to help you efficiently manage shipping and payment options.

## 5. Efficient Theme and Design Customization

Creating and maintaining individual themes for each store can be time-intensive, especially when aiming for distinct brand identities.

**Solution:** Optimize your theme management with a responsive and customizable theme framework. Reuse components and design elements across stores. Consider using Magento's built-in theme fallback system to inherit designs from a parent theme while making specific customizations for each store.

## 6. Streamlining Inventory Management

Managing inventory and stock levels across multiple stores, especially when shared products, can pose a challenge.

**Solution:** Simplify inventory management with specialized extensions. It lets you synchronize stock levels across your stores and provides real-time inventory tracking.

## 7. SSL Certificate Limitation

You may encounter a common SSL certificate issue when configuring multi-store functionality for your e-commerce website. It occurs because a single SSL certificate is assigned to one IP address and domain.

**Solution:** Two solutions to fix the SSL issue with your Magento 2 multiple sites:

1. Wildcard SSL: It provides security for all the subdomains you create, ensuring Comprehensive coverage
2. SAN (Subject Alternative Name) Certification: It is a solution for stores established on multiple domains, accommodating diverse configurations.

## 8. IP Address

There is a misconception that using a shared IP address negatively impacts an e-commerce store's online presence. Many well-known websites operate successfully with a single IP address, which does not necessarily hinder their online visibility.

## Magento Hosting Plan

Choosing a managed **Magento hosting** service is optional. You can choose a hosting service if you need:

- A website with optimised performance.
- Secure storage and bandwidth allocation.
- Automated backups.
- A scalable hosting solution.
- Expert support for hosting multiple sites.

Managed hosting can significantly improve the efficiency and reliability of your Magento multi-store setup. Evaluate your business needs to determine if managed hosting is the right choice for your Magento stores.

## Magento2 Components

### Hardware and Software Components Overview

**Web Server:** Magento 2 is compatible with Apache or NGINX web servers. Both are widely used and provide excellent performance and support for running a Magento 2 store.

**PHP:** Magento 2 requires PHP 7.4 or later, the scripting language used for server-side scripting and executing Magento code.

**MySQL:** To store data, Magento 2 needs a MySQL database server, version 5.7 or 8.0.

**Elasticsearch:** Magento 2 requires Elasticsearch 7.x for its powerful search capabilities and improved indexing performance.

## Authentication Keys

Before installing, you must obtain Magento authentication keys from the Magento Marketplace. These keys are essential for accessing Magento's repository to download and install the required software packages.

To obtain the authentication keys, create an account on the Magento Marketplace. Navigate to the "Access Keys" section, and generate a pair of public and private keys.

## File System Owner

Setting up the correct file system owner is crucial for the smooth operation of your Magento 2 store. The Magento file system owner manages all files and directories related to the installation.

It ensures the web server can access and modify the files when required. To set up the file system owner, follow these steps:

Create a new user on your server, for example, magento.

Add the new user to the web server group (www-data for Apache or nginx for NGINX). Change the ownership of the Magento 2 files and directories to the new user and web server's group.

## Example: Components Magento2

- **PHP:** Magento 2 requires PHP 7.x or later, leveraging the language's latest features for improved performance and security.
- **Database:** MySQL or MariaDB is used to store product data, customer information, order details, and other essential e-commerce data.
- **Elasticsearch:** An optional but recommended component for handling catalog search capabilities, providing fast and relevant search results.
- **Redis:** An advanced key-value store used for full-page cache (FPC) and session storage, significantly improving site performance.
- **Varnish:** A highly efficient HTTP accelerator and reverse proxy server that can be used to cache static content, further enhancing site speed.
- **Apache or Nginx:** Web servers compatible with Magento 2, each with its configuration specifics for serving web content.
- **Composer:** A dependency management tool for PHP, allowing for easy management of Magento 2 extensions and libraries.
- **RabbitMQ** (optional): A message queue system that helps with asynchronous processing, improving scalability and performance for complex operations.
- **Laminas (formerly Zend Framework):** A collection of PHP libraries used by Magento 2 for various functionalities, including input validation, filtering, and more.

## Part - A Questions

1. What is E-Commerce?
2. What is B2C ?
3. What is B2B ?
4. What is C2C ?
5. What is C2B ?
6. What is P2P ?
7. What is D2C ?
8. What is mobile commerce ?
9. What is social commerce ?
10. Define SaaS.
11. Define PaaS.
12. What is Magneto ?

13. Why to use Magento?
14. Define Xampp Server.
15. Define Composer function in Magento
16. Define Elasticsearch in Magento
17. What are the layers in Magento Framework?
18. List the Magento2 components

### **Part - B Questions**

1. How does E- Commerce work? Define with example.
2. Explain the types of E- commerce in real world.
3. Explain the Advantages and Disadvantage of E- Commerce.
4. Write a short notes about the basic features of Magento
5. Write a short notes about the Magento Folder Structure.
6. Explain the Magento Block Diagram
7. Explain the Global ,Website and Store methodology in Magento
8. Explain the Magento2 Components with example

### **Part - C Questions**

1. Explain the main types of E - Commerce business models
2. Describe the E- Commerce platforms with example
3. Explain the Advantages and Disadvantages of Magento
4. Explain the Magento version and features
5. Explain the Magento Architecture functionality.
6. Explain the Magento Frame Work Concepts.
7. Explain the Magento Design Concept and Terminology with scenarios
8. Explain the planning for multiple stores in Magento