

Overview of Platforms

Desktop

- ▀ Desktop programs are built for standalone computer desktop systems
- ▀ The builds are native to the underlying operating systems (OS). Typically,
 - Windows
 - Mac
 - Linux (to a lesser extent)
- ▀ In all cases, programs may be GUI enabled or not
 - Most user applications are GUI enabled.

Topic 2:

Overview of Platforms

32

- * Typically, applications software solutions are built with target platform in mind.
- * In this course, we well draw attention to the following platforms:
 - Desktop
 - Web
 - Mobile
 - Blockchain
 - Immersive Technologies

Topic 2:

...Overview of Platforms

...Desktop

- ▀ Each OS provider also provides SDK (Software Development Kit) that enables solutions to be developed for that environment
 - Windows SDK
 - See <https://developer.microsoft.com/en-us/windows/downloads/windows-sdk>
 - iOS SDK
 - See <https://developer.apple.com/ios/what-s-new/>
 - Typically bundled with Xcode IDE
 - Linux SDK
 - Each Linux distribution ships SDK. E.g.
 - Ubuntu:
<https://phone.docs.ubuntu.com/en/platform/sdk/>

Topic 2: ...Overview of Platforms

...Desktop



- Along with the SDK, each provider typically recommends some language(s) and IDE, as primary to the platform
 - Windows
 - IDE: Visual Studio (<https://visualstudio.microsoft.com/>)
 - Languages: C-sharp (C#), C++
 - Code samples: <https://github.com/microsoft/Windows-universal-samples>
 - iOS
 - IDE: Xcode (<https://developer.apple.com/xcode/>)
 - Linux
 - IDE: some basic ones like gedit ships with operating system. More matured ones like Eclipse are written in Java.

Topic 2:

...Overview of Platforms

...Desktop

- ❖ JavaScript has emerged as a language for building native applications for various OS platforms
- ❖ The respective SDKs are required, under the hood
 - <https://microsoft.github.io/react-native-windows/>
- ❖ We shall explore this approach to desktop application development in this course.

Topic 2: ...Overview of Platforms Web

- » **Web Stack** refers to the **layers of software used for implementing solutions** on the Web.
- » The most basic solution on the Web is the publication of static web pages on Web server for access from a Web client
 - » This simply requires a stack made up of a **Web server** and **Web client**.

Topic 2: ...Overview of Platforms

...Web

- ▀ Essentially, we can say that the Web was born with two primary layers of software:
 - ▀ **Web Server** used for publishing documents.
 - ▀ **Web Client** used for rendering published documents.

Topic 2: ...Overview of Platforms

...Web

- ☞ The **World Wide Web** (also simply referred to as the **Web**) is a system for the publication of documents **on the Internet**.
- ☞ The documents are usually in hypertext format written in **HTML** (**Hypertext Markup Language**).
- ☞ The **Hypertext** or **Hypermedia** format makes the Web **navigable**.

Topic 2:
*...Overview
of Platforms*
...Web

- ▀ **What Protocols are Involved in www?**
- ▀ **1. Internet Protocol (IP)**
 - ▀ **How** data packets are sent between nodes on the Internet. Each node is assigned a logical address (IP address).

Topic 2:
*...Overview
of Platforms*
...Web

- ▀ ...What Protocols are Involved in WWW?
- ▀ 2. Transmission Control Protocol (TCP)
 - ▀ How message delivery is layered on top of IP and guaranteed to arrive at logical address (IP address). Messages are sent in segments.
 - ▀ Adds multiplexing such that multiple programs can use the same IP.

Topic 2:
*...Overview
of Platforms*
...Web

- ▀ ...What Protocols are Involved in WWW?
- ▀ 3. Universal Datagram Protocol (UDP)
 - ▀ Some programs like streaming media programs, use UDP instead of TCP.
- ▀ 4. HTTP (**Hypertext Transport Protocol**) is how documents are accessed and transported on the Web

*Topic 2:
...Overview
of Platforms*

...Web

- ▀ **Software Systems involved in WWW**
- ▀ 1. **Web Server** (Used for publishing HTML documents) e.g. Nginx, Apache, Microsoft IIS, etc.
- ▀ 2. **Domain Name Servers (DNS)** – used for resolving IP of domain names
- ▀ 3. **Simple HTML Web pages or Web Applications** that generate HTML Web pages

Topic 2:
*...Overview
of Platforms*
...Web

- ❖ ...**Software Systems involved in WWW**
- ❖ **4. Web Clients** (Applications that communicate with Web Servers)
 - ❖ Web Browsers e.g. Firefox, Chrome, etc.
 - ❖ Mobile Hybrid Apps e.g. gidimo on Android.
- ❖ **5. URL (Uniform Resource Locator):** System for identifying and locating published documents

Topic 2:
*...Overview
of Platforms*

...Web

Step 1

- Specify the desired URL

Step 2

- Resolve IP Address

Step 3

- Do HTTP Request to IP Address

Step 4

- Send HTTP Response to Client from Server

▀ Classical Web Server Access
Steps from Web Client

...Overview of Platforms

...Web

Step 1

- Specify the desired URL

Step 2

- Resolve IP Address

Step 3

- Do HTTP Request to IP Address

Step 4

- Send HTTP Response to Client from Server

Topic 2:

Focus of Web Stack

Steps 3 and 4

In other words Web stack is particularly focused on **HTTP request - HTTP response cycle**

Topic 2:
*...Overview
of Platforms*
...Web

- ▀ **Software involved in HTTP request - HTTP response cycle**
 - ▀ For simple html pages, software simply boils down to Web Server and Web Client
 - ▀ For more complex solutions, Web Server is complimented by other Web application solutions which have their runtime environments.

Topic 2:

*...Overview
of Platforms*

...Web

- ▀ ...Software involved in HTTP request - HTTP response cycle
 - ▀ The Web Applications in their turn also can include some software layers.
 - ▀ At the bottom of the stack is typically a database system.

Topic 2:

...Overview of Platforms

...Web

■ **A Schematic View of Server side Web Stack (A Layered Architecture Approach)**

Web Server

- Receives requests payload (HTML, etc.) from client
- Serves static content directly
- For dynamic content, sends requests to Application layer for processing
 - Receives response from Web application and transmits to client

Web Application

- Processes requests based on business or application logic
 - Interacts with database layer for dynamic data reading and writing
 - Generates and passes response (HTML, etc.) to Web Server for transmission to client

Database Layer

- Persists dynamic data

Topic 2: ...Overview of Platforms

...Web

- ▀ **As a physical location**, a Web server is a machine that stores documents published on the Web and that also runs the requisite server-side software for managing access to such documents.
- ▀ **From a software angle**, Web server refers to the very software that handles the document location and access control using its HTTP server implementation.

Topic 2:

...Overview of Platforms

...Web

- ▀ In order of popularity, the two dominant Web server software are Apache, Nginx.
- ▀ Among these, Nginx is used by the greatest number of sites with a market share is about 34.1% as at 22 March 2024.
- ▀ See
<https://w3techs.com/technologies/comparison/ws-apache,ws-nginx>

Topic 2: *...Overview of Platforms*

...Web

- ▀ For developers, Nginx comes in handy for implementation and use in a development environment.
- ▀ It is lightweight and very easy to implement as a proxy server to other backend applications.
- ▀ Let's implement Nginx
- ▀ Reference manual:
 - ▀ *Manual-a webstack architecture primer.pdf* on the eLearning portal

Topic 2:

...Overview of Platforms

...Web

- ❖ Get to know the following from the reference material above, posted on the eLearning portal.
- ❖ Know how to install Nginx on
 - ❖ Linux (e.g. Ubuntu)*
 - ❖ Windows
 - ❖ Mac
- ❖ Installation from pre-built vs source
 - ❖ Why from source?
- ❖ Post installation configuration of Nginx
 - ❖ Creating named virtual hosts
 - ❖ Enabling TLS in Nginx for HTTPS Connection

Topic 2:

...Overview of Platforms

...Web

- ▀ I assume that *docker* program is up and running on your system
- ▀ To confirm once more, run *docker -v* command on a terminal. It should show you the docker version that is running on your system.
- ▀ For illustration of Nginx in a docker environment, let's take an “answer to question” approach.
 - I have shared on the eLearning portal, a zip file named *CSC202_2022set.zip*.

...Overview of Platforms

...Web

Topic 2:

Content

- ▀ *docker-compose.yml* in root folder

- ▀ An *nginx-primer* folder that contains various configuration files

- ▀ Within *conf.d* folder

- ▀ *Dockerfile*

Static content

- ▀ Within *html* folder

Topic 2:

...Overview of Platforms

...Web

- ▀ To run...
- ▀ On root terminal, execute the command *docker-compose up*.

- ▀ If there is no Nginx docker image already on your system, it will go online and get one, build and then startup the service.

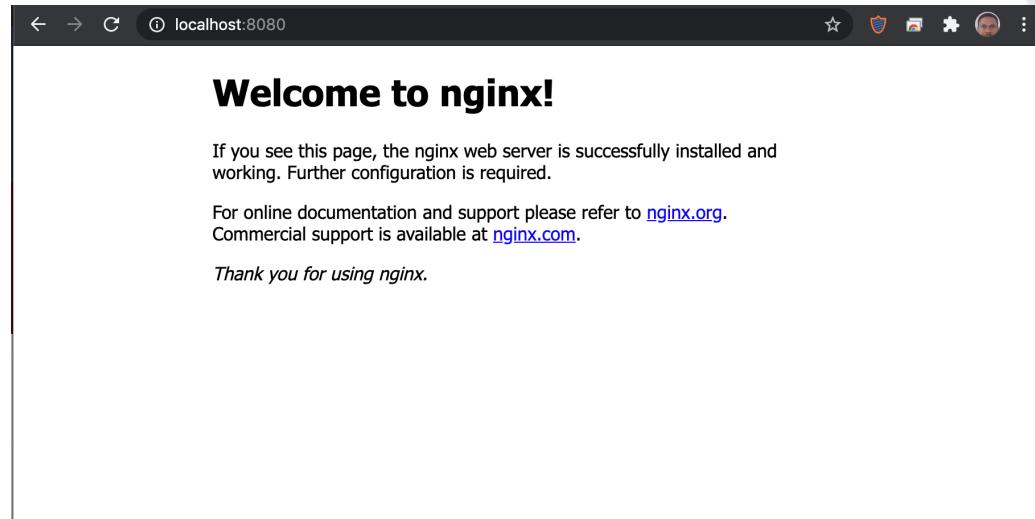
Topic 2:

...Overview of Platforms

...Web

56

- When nginx is running, you should see in browser from the URL <http://localhost>



Topic 2:

...Overview of Platforms

...Web

- ▀ Let's add a fake domain name *csc2022.com* to our local nameserver
 - ▀ For Mac
 - ▀ *sudo nano /etc/hosts*
 - ▀ Make the entry:
 - ▀ *127.0.0.1 csc2022.com*
 - ▀ *Ctrl-X* and allow save.
 - ▀ For Windows
 - ▀ Right-click Notepad program and select *Run as administrator*;
 - ▀ From Notepad, open the hosts file at:
C:\Windows\System32\drivers\etc\hosts
 - ▀ Make the entry:
 - ▀ *127.0.0.1 csc2022.com*
 - ▀ Save and close.

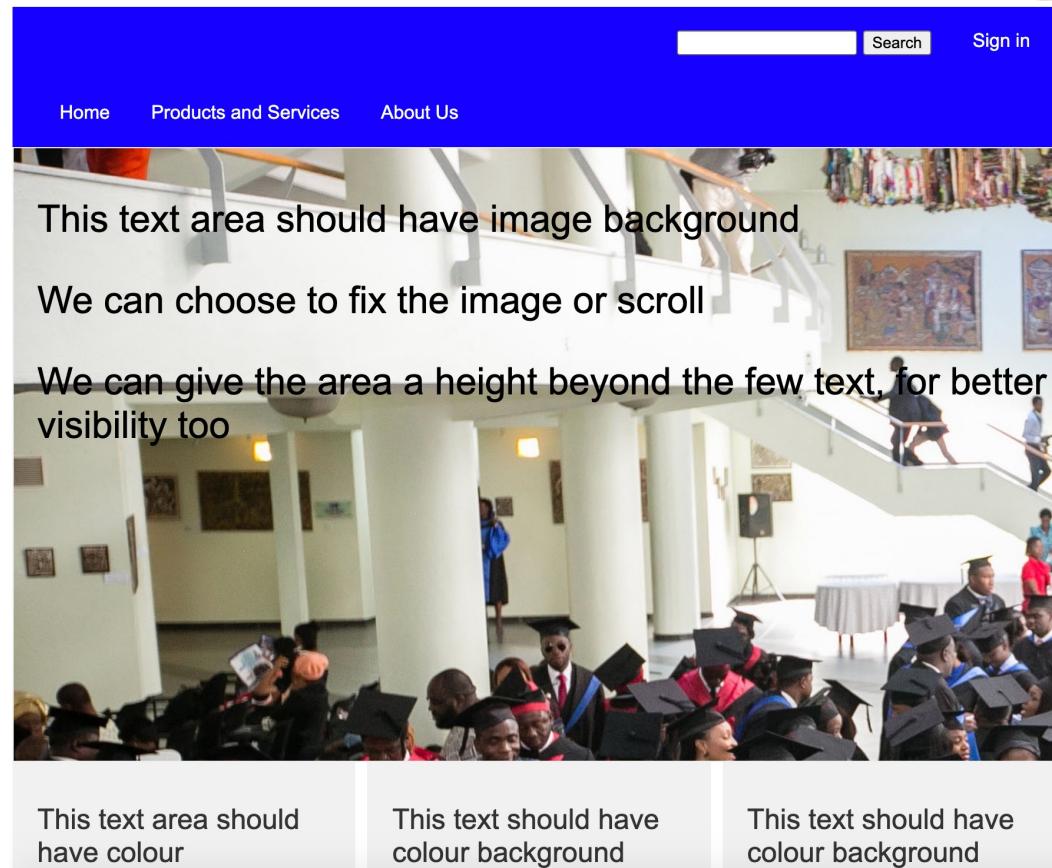
Topic 2:

...Overview of Platforms

...Web

58

- Now, you should see on your browser from the URL <http://csc2022.com/static-web-example.html>, the following view:



Topic 2: ...Overview of Platforms

...Web

- » Now that we have seen Nginx and docker in action, I believe that we should be curious to know what is going on, under the hood, who is responsible for what?

- » Time to listen to the facilitator in class!

Topic 2:

...Overview of Platforms

...Web

- ▀ With the docker implementation, we will assume in class that you have covered installation.
- ▀ We will jump to the section on “**Post installation configuration of Nginx**” in the manual named *Manual-a webstack architecture primer* on eLearning portal.
- ▀ Feel free to explore the direct installation on your own host operating system – Windows, Mac, Linux, as the case may be.

Topic 2: *...Overview of Platforms*

...Web

- ▀ Essentially, Nginx is a web server that checks for what to serve from the configuration file.
- ▀ Where is the configuration file?
 - Check with *nginx -V* command on terminal
 - By default, you will see
 - *--conf-path=/etc/nginx/nginx.conf*
 - How do you check filesystem on docker?
 - *docker exec -it nginx-csc202-2022set/bin/bash*
 - [Use *powershell* if your container contains Windows OS
 - *docker exec -it nginx-csc202-2022set powershell*]

...Overview of Platforms

...Web

Topic 2:

- Configuration of specific HTTP servers or virtual hosts

- Ref: page 10 of material

- Essential configuration structure:

```
http{
```

```
    server{
```

```
        server_name name-here e.g. csc2022.com;
```

```
        listen   port-here e.g. 80;
```

```
        location / {
```

```
            root    directory-here e.g. html;
```

```
            index   default-file-here e.g. index.html;
```

```
        }
```

```
    }
```

```
    server{
```

```
        ...
```

```
    }
```

include indicate where to read more server configurations from e.g. `/etc/nginx/conf.d/*.conf;`

```
}
```

Topic 2:
*...Overview
of Platforms*
...Web

- ▀ **Enabling TLS in Nginx for HTTPS Connection**
- ▀ Ref: page 13 of the reference manual
 - Follow the five steps

Topic 2: ...Overview of Platforms

...Web

- ❖ Dynamic Website in itself refers to that with dynamically generated content (e.g., from an underlying database) on HTTP request, as opposed to content already embedded in the published Web pages, on the server
- ❖ Same phenomenon applies to Web applications

Topic 2:

*...Overview
of Platforms*

...Web

- ▀ The difference however is
 - ▀ the term **Web application** is typically reserved for those with more complex or specialized dynamism logic (sometimes referred to as business logic)

Topic 2: ...Overview of Platforms ...Web

- ▀ For example,
 - ▀ Sites distinguished as social media sites (facebook.com, twitter.com, [linkedin.com](#)),
 - ▀ e-Commerce sites (e.g. [amazon.com](#)),
 - ▀ e-Learning sites (e.g. elearning.pau.edu.ng)
- ▀ are more appropriately referred to as Web applications.

Topic 2:

...Overview of Platforms

...Web

- ▀ On the contrary,
 - ▀ a website of an organization with dynamic information may simply be a **basic dynamic Website.**

Topic 2:

...Overview of Platforms

...Web

- ▀ For clarity of distinction, we will habitually qualify dynamic Website as basic when not referring to Web applications.
- ▀ In other words, we will use the terms **basic dynamic Websites** and **Web applications**.

Topic 2: ...Overview of Platforms

...Web

74

- ❖ The following are typical building blocks of dynamic Websites
 - ❖ 1. **Server side code** for
 - ❖ a. handling requests from client
 - ❖ b. implementing business logic
 - ❖ c. database access
 - ❖ 2. **Backend database**
 - ❖ 3. Optional **client side scripting** for interactivity e.g. form validation

Topic 2: ...Overview of Platforms

...Web

- » Server side code for basic dynamic Websites have often been implemented using PHP.
- » PHP was built as a server side programming language for the purpose. The acronym originally meant Personal Home Page.
- » It now refers to PHP Hypertext Preprocessor.

Topic 2: ...Overview of Platforms

...Web

- ▀ After installation on the server, the configuration approach depends on the Web server in use
 - ▀ For Apache, PHP is configured as an apache module.
 - ▀ For use in nginx, PHP processes are typically handled using another software e.g. PHP-FPM. FPM here means FastCGI Process Manager. CGI means Common Gateway Interface.

Topic 2:
*...Overview
of Platforms*
...Web

- ▀ Other languages are also used for server side scripting
 - ▀ ASP (Active Server Pages) by Microsoft and used in IIS (Internet Information Server) Web server environment
 - ▀ JSP (Java Server Pages) by SUN Microsystems (company bought by Oracle)
 - ▀ It requires a Servlet container like tomcat to run.

...Overview of Platforms

...Web

Topic 2:

- ❖ ...Other languages are also used for server side scripting
 - ❖ Python
 - ❖ Requires a Web server gateway software (e.g. wsgi) to use for server side code.
 - ❖ JavaScript
 - ❖ Originally desired for client side scripting
 - ❖ It has been ported for use on Server side in a NodeJS runtime environment.

Topic 2: ...Overview of Platforms

...Web

- ▀ For this course, we will start using Python but will highly encourage **JavaScript** for server side coding because of its versatility. JavaScript is now used for developing code for:
 - ▀ Web server side
 - ▀ Web client side
 - ▀ Native mobile app

Topic 2: ...Overview of Platforms

...Web

- ▀ So far, we have not seen any code examples in this topic.
- ▀ We shall do that after we have dealt with the concept of Web application frameworks.
 - ▀ dynamic Websites today are typically built on Web applications, and
 - ▀ Web applications today typically leverage on suitable pre-built Web application frameworks.

Topic 2:

...Overview of Platforms

...Web

Web Application Frameworks

- Web Application Framework (WAF) can also simply be referred to as Web Framework (WF).
- A WAF is a software designed to facilitate the development of web applications by **automating common activities** that are typically part of web development.
- WAFs have been developed for different languages
- Most of the WAFs use MVC (Model-View-Controller) Architecture

*Topic 2:
...Overview of
Platforms*

*...Web
Web
Application
Frameworks*

- ▀ Database Access
- ▀ Session Management. Typically HTTP session
- ▀ URL route definition and access control
- ▀ Template Engine or Template Processor. It facilitates the dynamic generation of HTML pages from templates and data

Topic 2:

...Overview of Platforms

...Web Examples of Web Application Frameworks in Different Languages

S/N	WAF Name + Project URL	Primary Language	Typical Template Engine	Popular Web App built with Framework
1	Flask + http://flask.pocoo.org/	Python	Jinja2	-Pinterest -Twilio
2*	Django + https://www.djangoproject.com/	Python	-Django Template Language (DTL) -Jinja2	-Instagram -Disqus -Pinterest -Washington Post -Eventbrite -Etc.
3	Zend + https://framework.zend.com/	PHP	-Zend view -PHP	-Magento eCommerce solution
4	Symfony + https://symfony.com/	PHP	-Twig -PHP	-Drupal Content Management System, Etc.
5	Laravel + https://laravel.com/	PHP	-PHP	-Mastercard -etc.

Topic 2:

...Overview of Platforms

...Web Examples of Web Application Frameworks in Different Languages

S/N	WAF Name + Project URL	Primary Language	Typical Template Engine	Popular Web App built with Framework
9.*	DotNet + https://www.microsoft.com/net/	C#	ASP.NET	-GTBank's online banking service -Microsoft site -Marketwatch -Etc.
10.	Express + http://expressjs.com/	JavaScript (Node.JS runtime env.)	-pug, hbs, nunjucks, etc	-Walmart -LinkedIn -Etc.
11.	Sail + http://sailsjs.com/	JavaScript (Built on Express)	-Embedded JavaScript (EJS) -CoffeeScript -TypeScript	-Jetblue -Verizon -Paystack -Etc.
12.	Koa + https://koajs.com/	JavaScript	-pug, hbs, nunjucks, etc	-Strapi -Brain hub -etc
13.	NestJS + https://nestjs.com/	JavaScript	-pug, hbs, nunjucks, etc	-Roche -Adidas -Scal.io -Autodesk, etc.

Topic 2:

...Overview of Platforms

...Web

Content Management Systems (CMS)

- Software designed to create and manage digital content.
- Web-based Content Management Systems (CMS) enable quick creation of dynamic web pages.
- Can ship with application modules out-of-the-box
- **Software coding not required for use out-of-the-box.**
- Less flexible for software coding than WAF.

Topic 2:
...Overview of Platforms
...Web
...Content Management Systems (CMS)

S/N	CMS Name + Project URL	Primary Language	Popular Web App built with CMS
1.*	Odoo + https://www.odoo.com/	Python	SIMS
2.*	ERPNext + https://erpnext.org/	Python	erpnext.com SaaS
2.*	Wordpress + https://wordpress.org/	PHP	PAU Website
3.	Liferay + https://www.liferay.com/	Java	UNAV.es
4.	DotNetNuke + http://www.dnnsoftware.com/	C# ASP	See http://www.dnnsoftware.com/community/participate/community-showcase
5.	-NodeJS CMSs + See https://www.slant.co/topics/1847/~node-js-based-cms for examples -Strapi – headless CMS (strapi.io)	JavaScript (NodeJS)	

Content Management Systems

Illustration with Odoo

- Content of *docker-compose.yml*

```
version: '3'
services:
  database:
    container_name: csc2022-set-odoo-db
    image: postgres:latest
    volumes:
      - /Users/piusonobhayedo/data/postgres/csc2022-set-odoo-db:/data/postgres
    environment:
      POSTGRES_USER: odoo
      POSTGRES_PASSWORD: odoo
      POSTGRES_DB: postgres
  odoo:
    container_name: odoo
    image: odoo
    links:
      - database:db
    ports:
      - 8069:8069
    depends_on:
      - database
```

Content Management Systems

*Illustration with Odoo
(cont'd)*

88

- ❖ Complete Odoo setup at the URL
<http://localhost:8069>



Warning, your Odoo database manager is not protected. To secure it, we have generated the following master password for it:

3fun-arf9-z2nt

You can change it below but be sure to remember it, it will be asked for future operations on databases.

Master Password

Database Name

Email

Password

Phone number

Language

Country

Demo data

Create database or restore a database

Content Management Systems

*Illustration with Odoo
(cont'd)*

89

▀ ...Complete Odoo setup at the URL <http://localhost:8069>



Warning, your Odoo database manager is not protected. To secure it, we have generated the following master password for it:

5jmy-4bph-iuqd

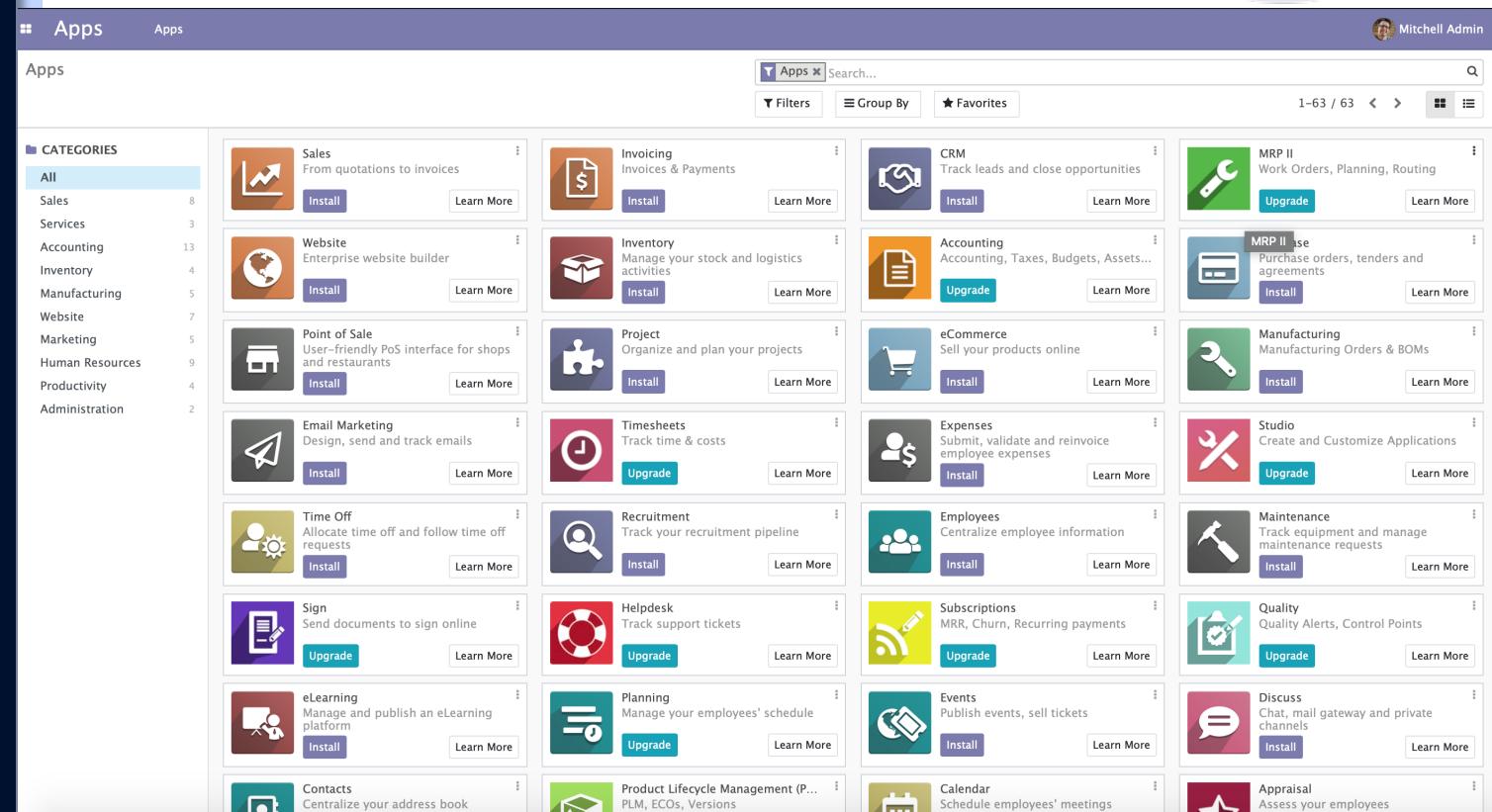
You can change it below but be sure to remember it, it will be asked for future operations on databases.

Master Password	<input type="password" value="*****"/>
Database Name	<input type="text" value="odoo"/>
Email	<input type="text" value="piosystems@yahoo.co.uk"/>
Password	<input type="password" value="*****"/>
Phone number	<input type="text" value="08175930381"/>
Language	<input type="text" value="English (US)"/>
Country	<input type="text" value="Nigeria"/>
Demo data	<input checked="" type="checkbox"/>
Create database or restore a database	

Content Management Systems

*Illustration
with Odoo
(cont'd)*

...Complete Odoo setup at the URL
<http://localhost:8069>



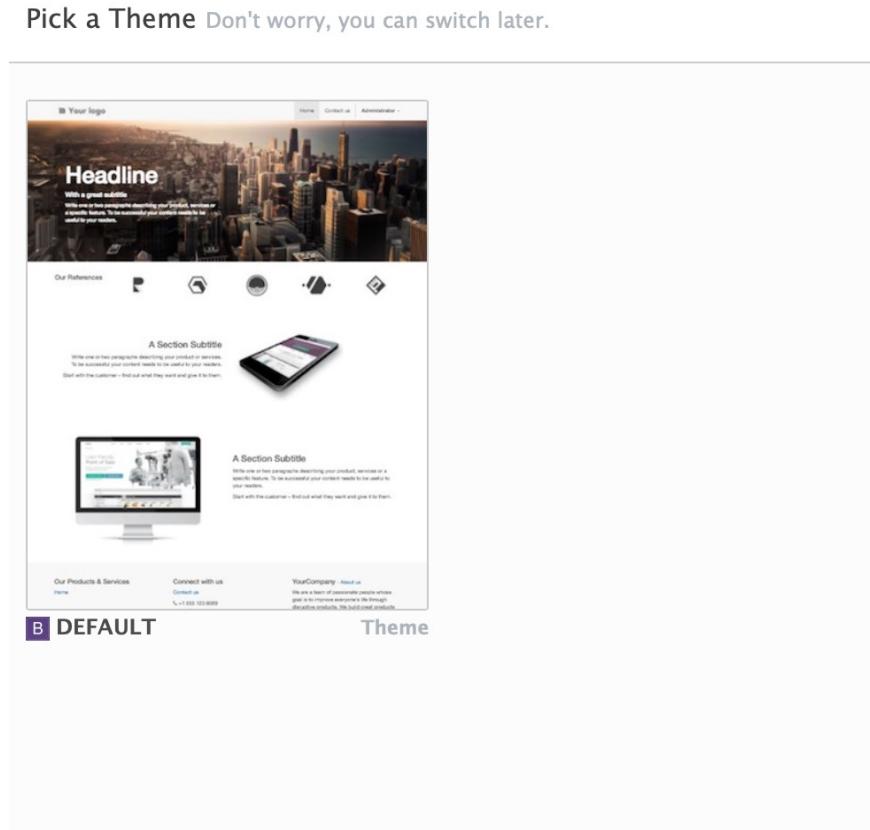
The screenshot shows the Odoo App Store interface. At the top, there's a search bar with 'Search...', filter buttons ('Filters', 'Group By', 'Favorites'), and a page indicator '1-63 / 63'. On the left, a sidebar titled 'CATEGORIES' lists 'All' and other categories: Sales (8), Services (3), Accounting (13), Inventory (4), Manufacturing (5), Website (7), Marketing (5), Human Resources (9), Productivity (4), and Administration (2). The main area displays a grid of 45 app cards, each with an icon, name, description, 'Install' or 'Upgrade' button, and a 'Learn More' link. The apps include Sales, Invoicing, CRM, MRP II, Website, Inventory, Accounting, Project, eCommerce, Manufacturing, Studio, Timesheets, Expenses, Recruitment, Employees, Maintenance, Sign, Subscriptions, Quality, eLearning, Planning, Events, Calendar, Product Lifecycle Management, and Appraisal.

Install desirable apps. You can begin with the app named *Website*

Content Management Systems

*Illustration with Odoo
(cont'd)*

- ▀ ...Complete Odoo setup at the URL <http://localhost:8069>
- ▀ Website app installation
 - Next window: Pick a theme



Content Management Systems

*Illustration with Odoo
(cont'd)*

92

- ▀ ...Complete Odoo setup at the URL <http://localhost:8069>
- ▀ Website app installation
 - Next window: Layout your page

The screenshot shows the Odoo website builder interface. At the top, there's a header with 'YOUR WEBSITE' and navigation links for 'Home' and 'Contact us'. On the right, it shows 'Mitchell Admin' and a 'Contact Us' button. Below the header is a large dashed rectangular area labeled 'DRAG BUILDING BLOCKS HERE'. To the right of this canvas is a sidebar titled 'BLOCKS' with sections for 'Structure', 'Style', and 'Options'. The 'Structure' section contains a grid of icons for various building blocks: Banner, Cover, Text - Image, Heading, Title, Text, Numbers, Picture, Columns, Big Boxes, Features, Masonry, Image Gallery, Images Wall, Carousel, Media List, Showcase, and Parallax. At the bottom of the sidebar, there are sections for 'Features', 'Comparisons', 'Team', and 'Call to Action'. The main content area below the canvas displays a dark-themed footer with sections for 'Useful Links', 'About us', 'Connect with us', and copyright information.

YOUR WEBSITE Home Contact us

Mitchell Admin Contact Us

DRAG BUILDING BLOCKS HERE

Blocks Style Options

Structure

Banner Cover Text - Image

Heading Title Text

Image - Text Picture Columns

Numbers

Big Boxes Features Masonry

Image Gallery Images Wall Carousel

Media List Showcase Parallax

Features

Comparisons Team Call to Action

Useful Links

Home About us Products Services Legal Contact us

About us

We are a team of passionate people whose goal is to improve everyone's life through disruptive products. We build great products to solve your business problems.

Connect with us

Contact us info@yourcompany.example.com 1 (650) 691-3277

Copyright © Company name English (US)

...Overview of Platforms

Mobile

Topic 2:

- ❖ Mobile Technologies can be studied from at least three perspectives
 - 1. Network
 - 2. Hardware
 - 3. Software
- ❖ Our primary focus is Software which can drive various services and solutions provided via mobile devices.
 - Discussion: Examples!
 - The nature of software supported depends on advancement of mobile technology which has varied across generations.

...Overview of Platforms

...Mobile

94

Topic 2:

- Mobile Technology Generations have been characterised by:
 - 1. Mobile Network's level of digital data bandwidth support
 - Remember 0G to 5G?
 - 2. Messaging Services Supported
 - Voice
 - SMS (Short Message Service)
 - USSD (Unstructured Supplementary Service Data)
 - 3. Robustness of installable applications (a.k.a mobile apps)
 - Dumb (feature phones)
 - Smart (Smartphones)
 - (Robustness of installable apps is usually dependent on
 - Computing capability and availability of advanced hardware features
 - Operating system maturity

Topic 2:

...Overview of Platforms

...Mobile Software

- ❖ Software has role to play not only in mobile apps, but also in voice services, SMS and USSD.
- ❖ In other words, voice, SMS and USSD are all programmable services.

- SMS is a one-way messaging system which supports store and forward).
- Short message service centre (SMSC) in mobile networks is used to store messages from a user for a period. The message is then forward to the destination user when the latter becomes available.
- A security disadvantage of SMS as a service is that unencrypted messages can be accessed by intruder with administrative rights to SMSC.

Topic 2:

...Overview of Platforms

...Mobile

...Software

...SMS Overview

- ❖ Every SMSC has an address which is a phone number. Mobile operators normally have theirs which is made available to the user.
- ❖ SMSC address is typically preset in sim card by mobile operators. So, user does not need to set it up for himself or herself.

Topic 2:

...Overview of Platforms

...Mobile

...Software

...SMS Overview

- ▀ SMSC can be implemented on a normal computer e.g. for bulk SMS messaging using a software application.
- ▀ Implementation tools examples
 - [https://www.jasminsms.com\)](https://www.jasminsms.com)
 - For docker based implementation, see
 - <https://jasmin.readthedocs.io/en/latest/installation/index.html#docker>
 - kannel.org

Topic 2:

...Overview of Platforms

...Mobile

...Software

USSD Overview

- ▀ USSD (Unstructured Supplementary Service Data) is a protocol for two-way communication between mobile phone and network operators' computers.
- ▀ Unlike SMS, it does not support store and forward.
- ▀ A USSD service is typically invoked by dialing a pre-assigned code preceded by * and ended with a #
 - ▀ e.g. *737#
- ▀ Normally implemented in collaboration with service providers.

Topic 2:

...Overview of Platforms

...Mobile

...Software

*...USSD
Overview*



- USSD Gateways are usually implemented by the mobile network operators (MNO) themselves
- To enable a USSD service, short code is required.
- Another software required on the server-side is the USSD application of the company engaged in marketing, that will process the USSD messages, following an established business logic.

- ❖ A number of development approaches
 - 1. Native development
 - A. Platform specific
 - B. Platform independent
 - 2. Hybrid app development
 - 3. Do It Yourself (DIY) tools

...Overview of Platforms

...Mobile

1A. Platform Specific Native Mobile App Development

102

Topic 2:



- Refers to the development of mobile app for use in a particular device or platform type running a specific operating system
 - iOS requires Objective-C or its successor, Swift programming languages. Development is typically done using Swift IDE (<https://developer.apple.com/swift/>)
 - Android requires Java. Android Development Toolkit (ADT) Plugin for Eclipse IDE is available from the Eclipse marketplace. Search for android.
 - Windows mobile (RIP) requires C#. Visual Studio IDE is typically used for native development.

Topic 2:

...Overview of Platforms

...Mobile

1B. Platform Independent Native Mobile App Development

- ❖ Refers to development of mobile app in such a way that native application can be generated from the same source code for different devices or operating system platforms.
- ❖ In other words, “*learn once, write anywhere*”.
- ❖ This approach requires suitable mobile development framework. They make mobile programming a little less tedious.
- ❖ **In this course, we'll mostly carry out mobile programming with such native code development approach.**

...Overview of Platforms

...Mobile

...1B. Platform Independent Native Mobile App Development

Topic 2:



Mobile development frameworks take care of routine complexities and save you time E.g.

- 1. Adding functionalities native to the mobile device like access to camera, GPS, etc.
- 2. Providing ready to use UI (User Interface) components rendered natively e.g. Text inputs, Buttons, List views, etc.



Examples of frameworks

- **React Native** (<https://reactnative.dev/>) for JavaScript developers
- **Flutter** (<https://flutter.dev/>) for Dart developers
- **Codename one** (<https://www.codenameone.com/>) for Java developers
- **Gluon** (<https://gluonhq.com/products/mobile/>) for Java developers
- **Xamarin** (<https://dotnet.microsoft.com/apps/xamarin>) for C# developers
- **In this course, we'll pay special attention to React Native.**

2. Hybrid Mobile App Development

...Overview of Platforms

...Mobile

Topic 2:

- ❖ Involves the use of suitable framework that enables combination of native application and Web application, as mobile application.
- ❖ Example
 - Apache Cordova. Programming is in JavaScript, HTML and CSS.
 - Ionic which combines Apache Cordova and angular js
- ❖ The mobile apps created in this way are less efficient than pure native mobile applications. Hence, we will not make this our focus.

Topic 2:

...Overview of Platforms

...Mobile

3. DIY (Do It Yourself) Mobile App Development

- ❖ Involves the use of visual tools requiring no programming skills. They are usually fast but less flexible.
- ❖ Mostly cloud-based and typically requires paid subscription.
- ❖ Let's take a look at the examples in the URLs
<https://www.websitetooltester.com/en/blog/app-makers/> and
<http://noeticforce.com/best-online-diy-mobile-app-builder-platforms>

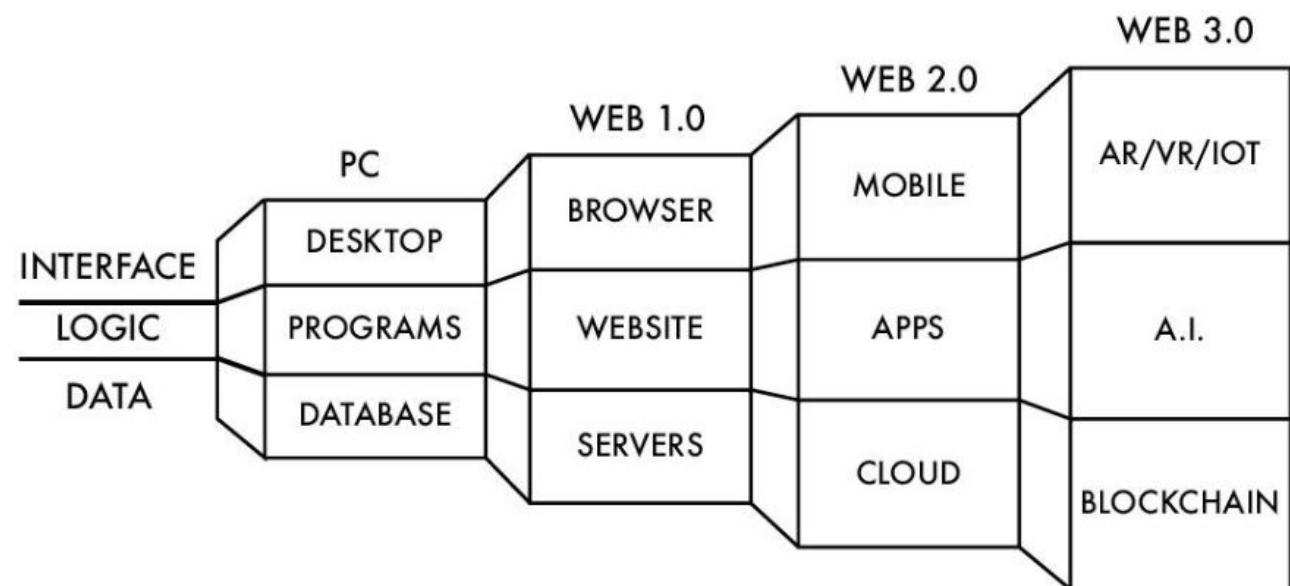
...Overview of Platforms

Blockchain

Topic 2:

Blockchain

– Evolution:



Topic 2:

...Overview of Platforms

...Blockchain

- ▀ Blockchain is distributed database of transaction records linked as blocks.
 - Immutable records
 - You can write data to and read from, but you can't delete or modify.
- ▀ Third party not required to verify transactions (*Trustlessness*)
 - Consensus algorithm does the job
- ▀ Solutions stack are as shown in the next page

Layer 4:

-dApps

1. Protocols ((e.g. DeFi like Automated Market Markers like <https://kyber.network/>, <https://uniswap.org/>, <https://balancer.fi/>,)),

2. Tokens

***fungible and non-fungible:** An nft cannot be replicated nor exchanged for another nft as if equivalent because no two nfts are the same. Nfts can thus be used as basis for tokenizing real world tangible assets as well as used as individuals' identity, property title, etc.

*Utility vs Payment vs Security/Asset vs Equity

3. Business logic -Available on:

1. non-Spatial (Mobile, Web)
2. Spatial (a.k.a. Metaverse - e.g. decentraland)

Layer 3:

Developer APIs

- | | |
|---------------------------------------------------------------------------------------------------------|-----------------------------------|
| 1) web3.js (see https://www.npmjs.com/package/web3) | 2) ether.js |
| 3) polkadot.js (see https://polkadot.js.org/docs/api/) | 4) Solidity (for smart contracts) |
| 5) Rust-based apis (for smart contracts and more) | |

Layer 2:

Meta-protocols for the enhancement for the lower layers e.g. in the area of scaling (e.g. Polygon protocol layered on Ethereum. See <https://polygon.technology/>) and security

Layer 1: Blockchain and Decentralized Oracle Networks. Oracle connects real-world data to blockchain

1. **Data distribution protocols** e.g. ipfs (interplanetary file system. Peer-to-peer fs. See ipfs.io)
2. Trustless **interaction platforms** e.g. polkadot
3. Trustless **interaction protocols** e.g. bitcoin, ethereum, polkadot parachains (**Blockchain properly speaking, with native coin**)
4. **Messaging protocols** e.g. matrix (see <https://matrix.org/>)
5. **Frameworks** like substrate

Layer 0:

1. Peer-to-peer (P2P) as Internet overlay

[Commonly used library - <https://libp2p.io/>.]

You should treat this in the course on net-centric computing. I highly recommend Rust implementation (<https://github.com/libp2p/rust-libp2p!>)]

2. Platform neutral computation description language e.g. **WASM (Web Assembly)**, EVM (Ethereum Virtual Machine), UTXO (Unspent Transaction Outputs. Used by BitCoin)

...

Layer 1:

1. **Data distribution protocols** e.g. ipfs (interplanetary file system. Peer-to-peer fs. See ipfs.io)
2. Trustless **interaction platforms** e.g. polkadot
3. Trustless **interaction protocols** e.g. bitcoin, ethereum, polkadot parachains (**Blockchain properly speaking, with native coin**)
4. **Messaging protocols** e.g. matrix (see <https://matrix.org/>)
5. **Frameworks** like substrate

Layer 0:

1. **Peer-to-peer (P2P) as Internet overlay**

[Commonly used library - <https://libp2p.io/>.

You should treat this in the course on net-centric computing. I highly recommend Rust implementation (<https://github.com/libp2p/rust-libp2p>!)]

2. **Platform neutral computation description language** e.g. **WASM (Web Assembly)**, EVM (Ethereum Virtual Machine), UTXO (Unspent Transaction Outputs. Used by BitCoin)

Internet

Layer 3:

Developer APIs

- 1) web3.js (see <https://www.npmjs.com/package/web3>)
- 2) ether.js
- 3) polkadot.js (see <https://polkadot.js.org/docs/api/>)
(for smart contracts)
- 4) Solidity
- 5) Rust-based apis (for smart contracts and more)

Layer 2:

Meta-protocols for the enhancement for the lower layers e.g. in the area of scaling (e.g. Polygon protocol layered on Ethereum. See <https://polygon.technology/>) and security

...

Layer 4:

-dApps

1. Protocols (e.g. DeFi like Automated Market Markers like <https://kyber.network/>, <https://uniswap.org/>, <https://balancer.fi/>,),

2. Tokens

***fungible and non-fungible:** An nft cannot be replicated nor exchanged for another nft as if equivalent because no two nfts are the same. Nfts can thus be used as basis for tokenizing real world tangible assets as well as used as individuals identity, property title, etc.

***Utility vs Payment vs Security/Asset vs Equity**

3. Business logic -Available on:

1. non-Spatial (Mobile, Web) 2. Spatial (a.k.a. Metaverse - e.g. decentraland)

...

...Overview of Platforms

...Blockchain

Topic 2:



- Emerging concepts (for discussion).
E.g.
 - Cryptocurrencies
 - Tokens
 - Non-fungible Tokens (NFT)
 - Fungible Tokens
 - Decentralized finance (DeFi)
 - Crypto Market making / Automated Market Making (AMM)
 - Decentralized Autonomous Organisations (DAO)
 - Smart contracts

...Overview of Platforms

Immersive Technologies

Topic 2:

- XR (AR/VR/MR) Concept
- Technology Stacks
 - Oculus Quest
 - HoloLens
 - WebXR
- FTs, NFTs and the Metaverse
 - Read
<https://www.siliconrepublic.com/business/metaverse-hype-trends-artificial-virtual-reality-2023>
 - <https://www.pewresearch.org/internet/2022/06/30/the-metaverse-in-2040/>