

INFS3202/7202 – Web Information Systems

Lecture Week 1: Course Overview and Intro to the WWW

Dr Aneesha Bakharia (Senior Lecturer, EECS)
Email: a.bakharia1@uq.edu.au

Acknowledgement of Country

The University of Queensland (UQ) acknowledges the Traditional Owners and their custodianship of the lands on which we meet.

We pay our respects to their Ancestors and their descendants, who continue cultural and spiritual connections to Country.

We recognise their valuable contributions to Australian and global society.



Contents

01 About the Lecturer and Course Coordinator

02 The Course Demonstrators

03 Course Overview

04 Assessment

05 Introduction to the World Wide Web

06 Introduction to Server-Side Web Programming

About Me – Dr Aneesha Bakharia

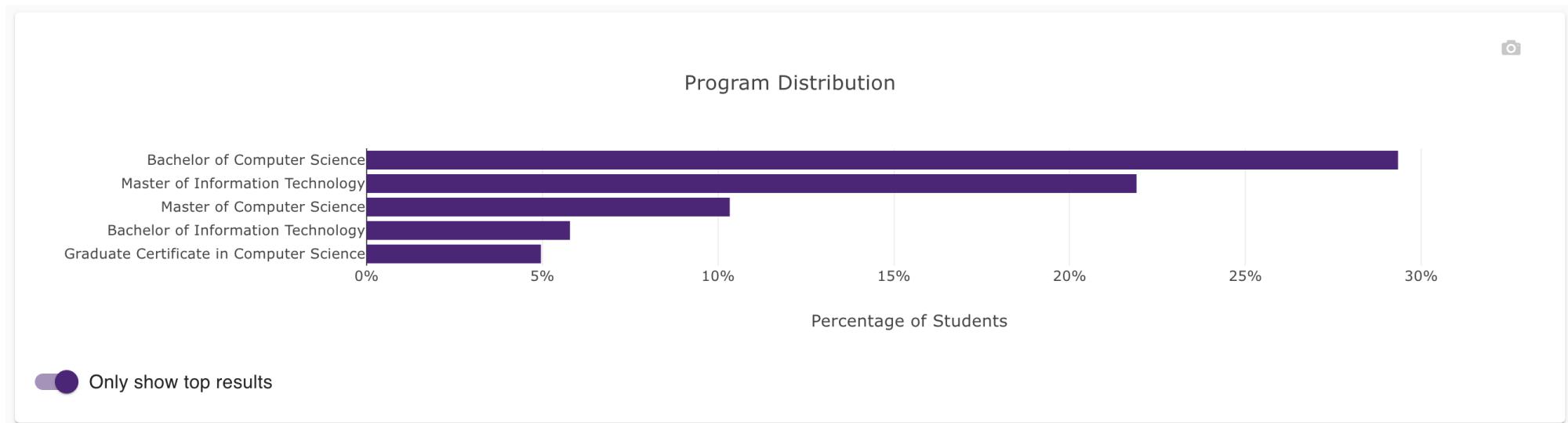
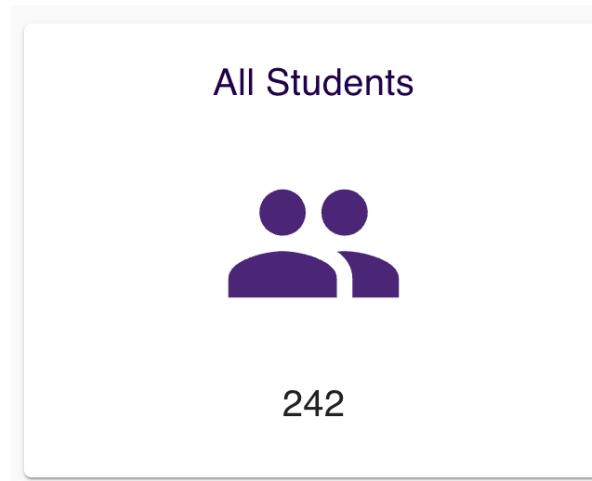
Lecturer and Course Coordinator for INFS3202/7202

- **Passionate about Web development and Generative AI**
- **PhD in Automated Text Analytics (QUT)**
- **Written 9 books on Web development and programming + 1 interactive tutorial for Manning (book publisher)**
- **Worked at UQ (7 years) as the Manager of Learning Analytics**
- **Now full-time Senior Lecturer in EECS**
- **Active on social media: @aneesha on X (Twitter)**

Course Demonstrators

- Mr Spencer Kayser (Lead Demonstrator)
- Mr Zaidul Alam
- Mr Blake De Raat
- Mr Frank Thurnbacher
- Ms Jasmine Burt
- Mr Joseph Krebs
- Mr Kwan Li
- Mr Firlandi Ansyari
- Mr Bach Truong

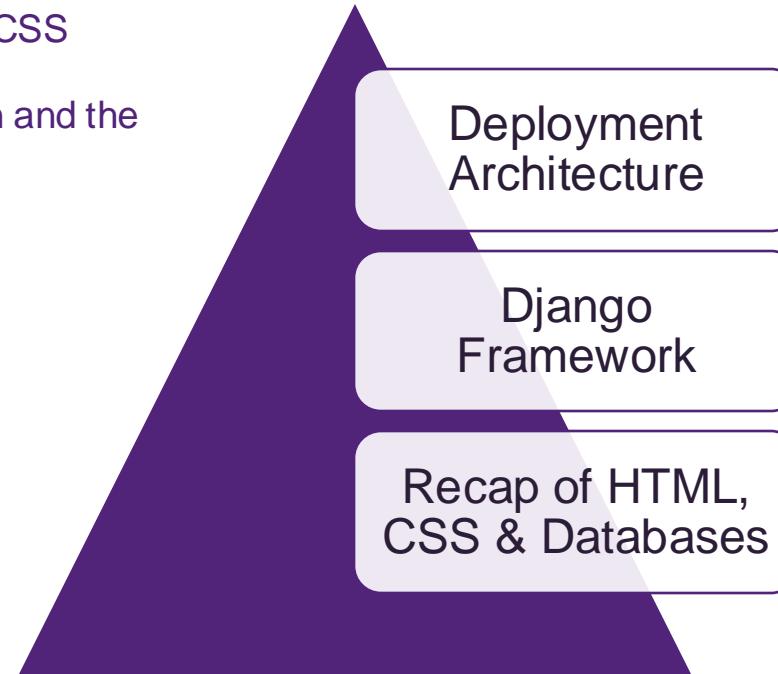
Who are the Students taking INFS3202/7202?



INFS3202/7202 – What will you learn?

Building Web Applications

- Foundational web technologies HTML and CSS
- Server-side web development using Python and the Django Framework
- Designing databases for Web applications (including for searching)
- Designing web UI's using CSS and HTML Forms
- Incorporating GenAI into Web applications (including RAG)
- Deploying Web Applications (via Nginx Web server and AWS)



Building UI's for Smart App built on Gen AI

← Back to Templates

Liftoff – AI Mock Interview Simulator

Liftoff is an interview preparation tool that provides AI feedback on your mock interviews, built with OpenAI Whisper and GPT.

Framework

Next.js

Use Case

AI

CSS

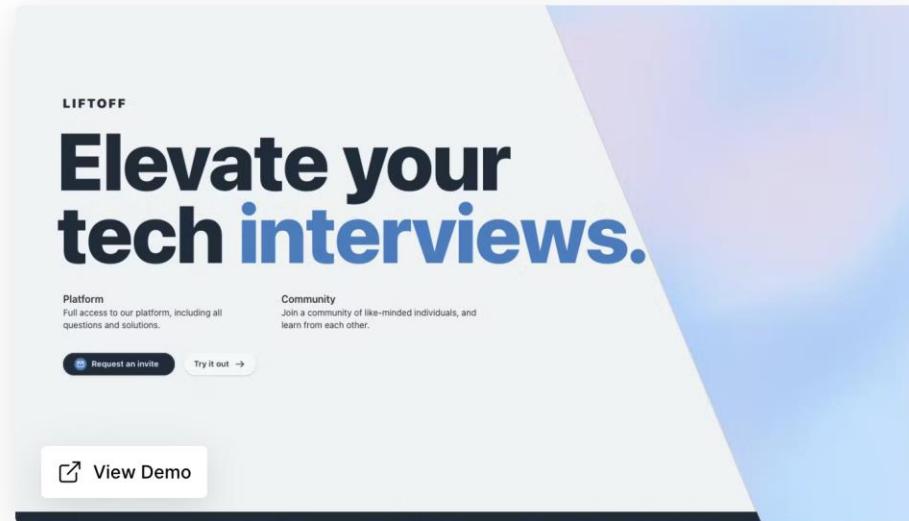
Tailwind

Publisher

Tyler Meyer

Deploy

View Repo



Introduction

Liftoff is an interview preparation tool that provides AI feedback on your mock interviews.

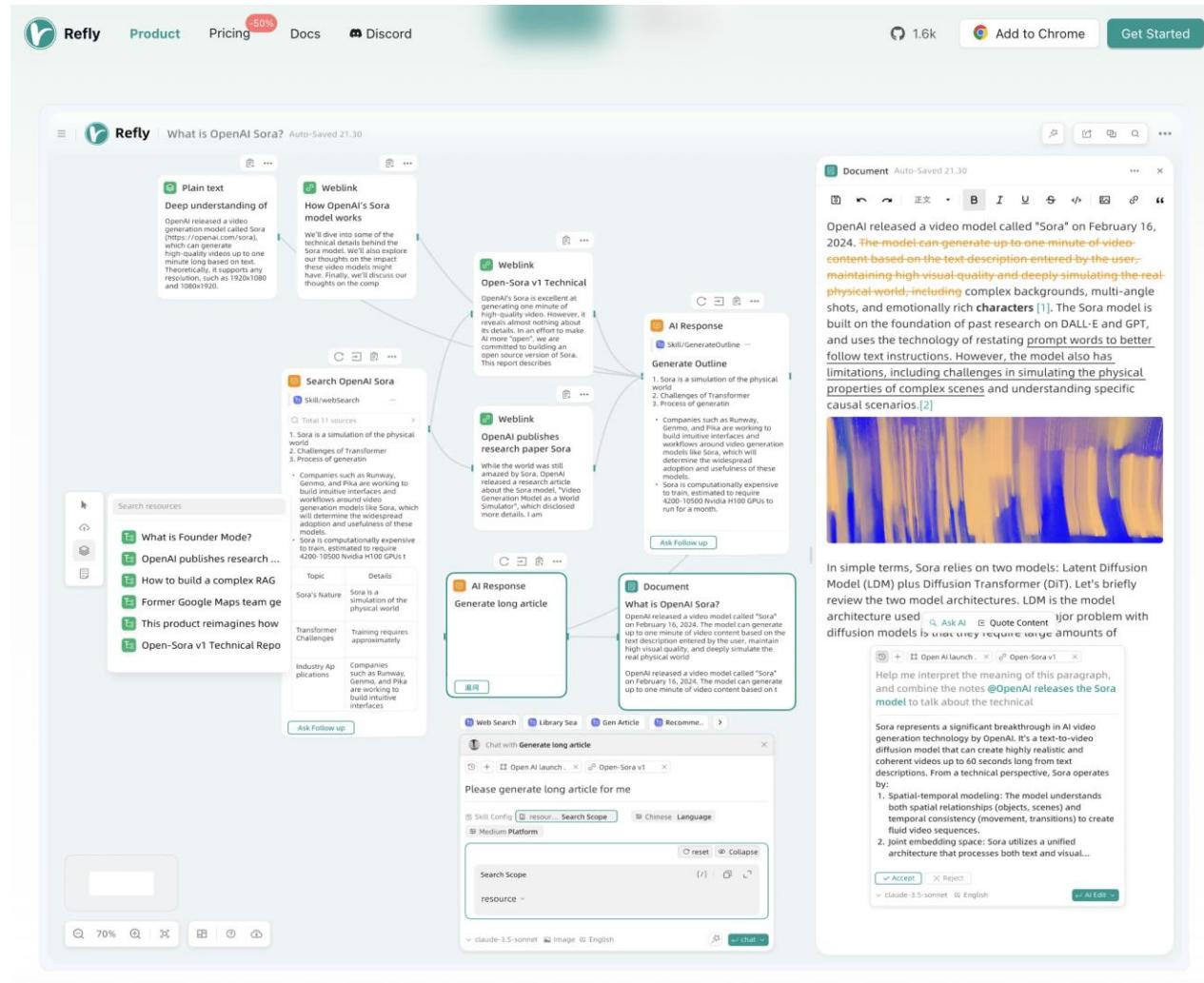
One-click Deploy

You can deploy this template to Vercel with the button below:

You can also clone & create this repo locally with the following command:

<https://vercel.com/templates/next.js/liftoff>

Building UI's for Smart App built on Gen AI



<https://refly.ai/>

Learning Objectives

From Course Profile

1. Apply system architecture principles to design and deploy Web Information Systems (WIS) solutions.
2. Evaluate and articulate the scope, complexity, and key considerations in the design and implementation of Web Information Systems.
3. Design and program Web Information Systems (WIS) with server-side functionalities.
4. Develop responsive Web-based, database-driven applications using efficient and effective technologies.
5. Evaluate and justify the suitability of Web Information Systems solutions in various contexts, considering factors such as user needs and technical constraints.
6. Judge in which situations WIS solutions are more or less appropriate.
7. Critically analyze current issues and emerging trends in Web Information Systems development, and predict potential impacts on future practices and technologies.

Course Pre-requisites

- Web development has a lot of pre-requisites
- One Week Self Paced Guides Provided
 - HTML
 - CSS
- An introduction to databases and Python will be covered in lectures

Orientation Week (O Week)

Build Content  Assessments  Tools  Partner Content 



Programming Background Survey

Please take a moment to fill out this short survey on your programming background. The data will be used to better customise the course material and practicals to your unique learning needs and programming background.

[Take the Programming Background Survey](#)



Self-Regulated Learning Questionnaire

Enabled: Statistics Tracking

The SRL-O questionnaire is a tool for you to check how well you're managing your studies and get feedback. It looks at your study habits, how motivated you are, and how you handle challenges while learning. By answering its questions, you'll get personalized tips on how to study better and make the most of this course.

[Take the Self-Regulated Learning Questionnaire](#)



Introduction to HTML Tutorial (Optional)

Attached Files:  [Introduction to HTML Tutorial](#)  (1.135 MB)
 [Solution](#)  (1.286 KB)



Introduction to CSS (Optional)

Attached Files:  [Introduction to CSS Tutorial](#)  (1.111 MB)
 [Solution](#)  (2.59 KB)

Lectures and Labs

- **2 hour Lecture on Thursday**
- **2 hour Lab (on either Mon, Tues or Wed)**
- **Lab content is 1 week behind the Lecture**

	Mon	Tue	Wed	Thu
8:00 AM	INFS3202_S1_STLUC_IN PRA1 01 3/3-14/4, 28/4-26/5 78-116 - General Purpose South, Computer Lab	INFS3202_S1_STLUC_IN PRA1 05 4/3-15/4, 29/4-27/5 78-116 - General Purpose South, Computer Lab		
9:00 AM				
10:00 AM	INFS3202_S1_STLUC_IN PRA1 02 3/3-14/4, 28/4-26/5 78-116 - General Purpose South, Computer Lab		INFS3202_S1_STLUC_IN PRA1 08 5/3-16/4, 30/4-28/5 78-116 - General Purpose South, Computer Lab	
11:00 AM				
12:00 PM	INFS3202_S1_STLUC_IN PRA1 03 3/3-14/4, 28/4-26/5 78-116 - General Purpose South, Computer Lab	INFS3202_S1_STLUC_IN PRA1 06 4/3-15/4, 29/4-27/5 78-116 - General Purpose South, Computer Lab		
1:00 PM				
2:00 PM		INFS3202_S1_STLUC_IN PRA1 07 4/3-15/4, 29/4-27/5 78-116 - General Purpose South, Computer Lab	INFS3202_S1_STLUC_IN PRA1 09 5/3-16/4, 30/4-28/5 78-116 - General Purpose South, Computer Lab	INFS3202_S1_STLUC_IN PRA1 01 27/2-17/4, 1/5-29/5 08-139 - Goddard Building, Learning
3:00 PM				
4:00 PM	INFS3202_S1_STLUC_IN PRA1 04 3/3-14/4, 28/4-26/5 78-116 - General Purpose South, Computer Lab			INFS3202_S1_STLUC_IN PRA1 01_Delayed 27/2-17/4, 1/5-29/5 Delayed viewing (Clashable)
5:00 PM				
6:00 PM				

Lectures and Practicals – Week 1 to 6

Week	Lecture	Practical	
Week 1	Course Overview & Intro to WIS	No Lab in Week 1	
Week 2	Django MVC Framework Introduction (includes creating base templates, basic routing and views)	Lab 1: UQCloud, Nginx & VSCode Online	
Week 3	MVC 1 – Models & SQL Databases (includes databases recap, Django Object Relational Models (ORM) and MySQL)	Lab 2: Your First Django Project	Lab 2 and 3 are challenging. We'll include recaps and they follow lecture content.
Week 4	MVC 2 – Routes and Views (includes UX prototyping with CSS libraries)	Lab 3: Databases and Models	Ask questions, the Teaching Team are here to support your learning.
Week 5	MVC 3 – Creating CRUD Applications (Includes Full Text Search)	Lab 4: Designing UI's with CSS Frameworks	
Week 6	MVC 4 – Advanced topics (File uploads, caching, sessions, authentication & authorisation and testing)	Lab 5: Work on Design Document Design Document Due 3pm Friday 4 Apr	

Lectures and Practicals – Week 7 to 13

Week 7	Incorporating GenAI features in Web Applications (Calling GenAI API's, Creating Chatbots and Retrieval Augmented Generation using LangChain)	Lab 6: Login and Form Processing
Week 8	Designing RESTFul API's & Using Client-side Javascript (Django REST Framework)	Lab 7: Incorporating GenAI
Mid-Semester Break		
Week 9	Developing Progressive Web Applications and Web Accessibility (Responsive CSS, PWA's & Accessibility)	Lab 8: Creating RESTFul API's
Week 10	Deploying to the Cloud - Guest Lecturer from AWS	Lab 9: Work on Project
Week 11	Web Security	Lab 10: Deployment to AWS Project Due 3pm Friday 16 May 2025
Week 12	Other Web frameworks (Flask, FastAPI, FastHTML, Next.JS)	Lab 11: Project Demonstration
Week 13	Revision	Lab 12: Exam Revision

Assessment

From Course Profile: <https://course-profiles.uq.edu.au/course-profiles/INFS3202-21297-7520#assessment>

Assessment summary

Category	Assessment task	Weight	Due date
Computer Code, Participation/ Student contribution	<u>Weekly Activities</u>  Online	15%	7/03/2025 - 16/05/2025 Due weekly at 3pm on Friday from Week 2 - 11, except in Week 8 where the due date is Thurs 17/4/25.
Paper/ Report/ Annotation, Project	<u>Design Document</u>	20%	4/04/2025 3:00 pm Due at the end of Week 6.
Computer Code, Project	<u>Web Project</u>  In-person	30%	16/05/2025 3:00 pm
Examination	<u>Final Exam</u>  Hurdle  Identity Verified  In-person	35%	End of Semester Exam Period 7/06/2025 - 21/06/2025

- You can use Gen AI but will need to reference your use of Gen AI in your assessment submission.

Weekly Activities

From Course Profile: <https://course-profiles.uq.edu.au/course-profiles/INFS3202-21297-7520#assessment-detail-0>

Weekly Activities



Online

Mode

Written

Category

Computer Code, Participation/ Student contribution

Weight

15%

Due date

7/03/2025 - 16/05/2025

Due weekly at 3pm on Friday from Week 2 - 11, except in Week 8 where the due date is Thurs 17/4/25.

Other conditions

Peer assessed.

[See the conditions definitions](#)

Learning

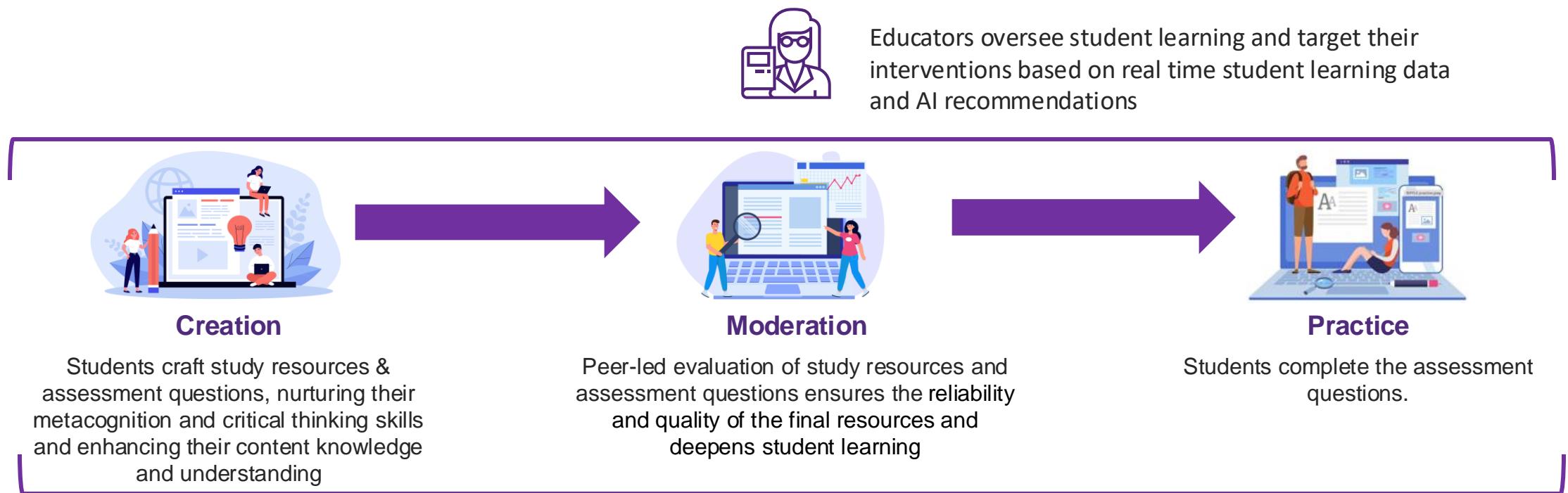
outcomes

L01, L02, L03, L04, L05, L07

Task description

In the Weekly Activities assessment, students will regularly participate in a series of activities via the course's Blackboard site. These activities will enable students to track their learning and received regular feedback through regular quizzes with peer interaction.

RiPPLE uses the science of learning and AI to help students deepen and demonstrate their learning



Throughout the process, AI provides students with pedagogical support that time-limited instructors can't provide

Project Options

Design Document and Web Project Assessment Items

Project Name	Description
TableTap	TableTap is a Software as a Service (SaaS) platform designed specifically for restaurants, cafes, and coffee shops to streamline their ordering process.
InsightHub	InsightHub is a Software as a Service (SaaS) platform designed to help users make dashboards. It allows users to sign up and upload CSV files, which can contain any type of data they wish to visualize.
LearnMore	LearnMore is a Software as a Service (SaaS) platform designed to simplify the process of building online courses.
Custom (Your Idea)	You can submit a proposal for a project idea for a SaaS platform. You must submit your proposal to the Course Coordinator via email by the end of Week 2. You can decide if you want to make the project idea available to other students.

- The web project must be implemented in either Django or Next.js. The course teaches Django and students already familiar with React can choose to learn Next.js on their own. Note that some of the questions in the Exam will be related to Python and Django.

Design Document – Assessment Item

From Course Profile

Design Document

Mode Product/ Artefact/ Multimedia, Written

Category Paper/ Report/ Annotation, Project

Weight 20%

Due date 4/04/2025 3:00 pm

Due at the end of Week 6.

Learning outcomes L01, L02, L05, L06, L07

Task description

1. The assignment is designed to test your ability to design and document a WIS, with a focus on the database and web user interface design.
2. Students will be able to select from one of three project topics. Students will be required to select ONLY ONE of the projects and author a comprehensive design document.
3. For the students who have extensive experience in Web system development, you have the option to propose a new project topic. However, you must submit a project proposal in Week 2 and seek pre-approval from the Course Coordinator.

Project – Assessment Item

From Course Profile

Web Project



In-person

Mode	Oral, Product/ Artefact/ Multimedia
Category	Computer Code, Project
Weight	30%
Due date	16/05/2025 3:00 pm
Learning outcomes	L03, L04

Task description

1. The assignment is designed to test your ability to develop a WIS based on a specification and design document, with a focus on the technology part of this course.
2. The implemented project must match your submitted Design Document. Students will not be allowed to change project topics.
3. This assessment item has two parts:
 - Part one is an online code submission. All code will be required to be compressed into one file and submitted via Blackboard in Week 11 (3pm Friday). The submission will also need to include a link to the online deployment of the web project to UQCloud.
 - Part two is an oral demonstration and review of your submitted code, which will be completed during an in-person meeting with a demonstrator scheduled during the Lab session in Week 12, after the code submission. In the meeting with the demonstrator, students will demonstrate all implemented functionality, explain their approach and answer questions specific to their code implementation. All oral assessments must be given live and will be recorded by the teaching team for archiving purposes.

Exam – Assessment Item

From Course Profile

Final Exam

 Hurdle  Identity Verified  In-person

Mode	Written
Category	Examination
Weight	35%
Due date	End of Semester Exam Period
7/06/2025 - 21/06/2025	
Learning outcomes	L01, L02, L03, L04, L05, L07

Task description

This course will have one final exam that is designed to test theoretical concepts and programming skills introduced in this course. The exam will address all materials covered by lectures and labs.

Hurdle requirements

The exam is an Identity Verified Assessment Hurdle. If the exam is not passed with a minimum score of 17.5/35, the final grade received will be capped at a 3 even if the overall score received is of greater than or equal to 50%.

- Identify Verified and a Course Hurdle
(i.e. You must pass the Exam to Pass the course)

Plagiarism

- Plagiarism is the act of misrepresenting as one's own original work the ideas, interpretations, words or creative works of another. These include published and unpublished documents, designs, music, sounds, images, photographs, computer codes and ideas gained through working in a group. These ideas, interpretations, words or works may be found in print and/or electronic media.
- Students are encouraged to read the UQ Student Integrity and Misconduct policy (<http://ppl.app.uq.edu.au/content/3.60.04-student-integrity-and-misconduct>) which makes a comprehensive statement about the University's approach to plagiarism, including the approved use of plagiarism detection software, the consequences of plagiarism and the principles associated with preventing plagiarism.
- All submitted works will be tested with an electronic plagiarism check
- Code between student submissions will also be checked for similarity

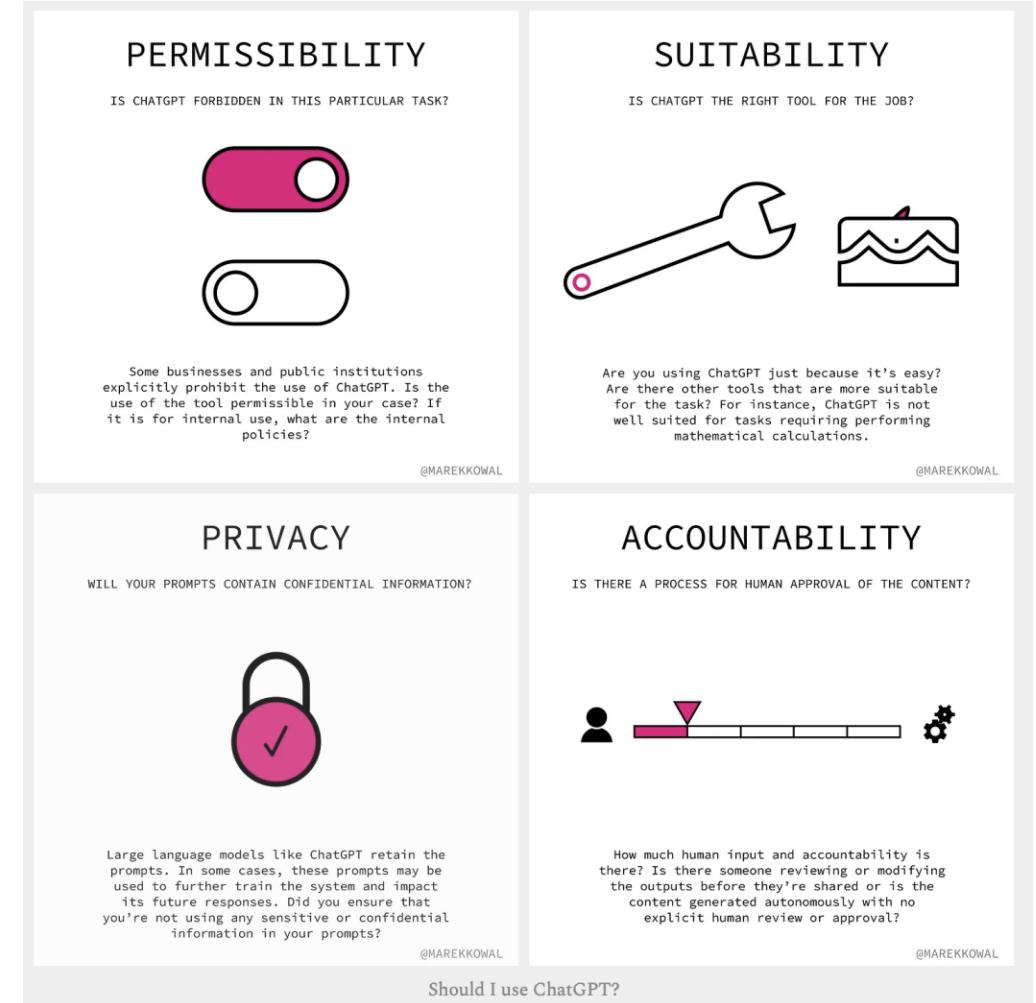
Access the Academic Integrity module

1. Go to learn.uq.edu.au
 2. Click on the Academic Integrity Module link

The screenshot shows a university portal interface. On the left, a sidebar titled 'Tools' lists various links: Announcements, Calendar, Tasks, My Grades, User Directory, Address Book, Personal Information, My OHS Results, My Media, Application Authorization, Academic Integrity Module, iThenticate, my ePortfolio, and my ePortfolio account setup. The 'Academic Integrity Module' link is circled in red with a large red arrow pointing to it from the bottom left. The main content area is titled 'My Courses' and displays a list of courses, many of which are blurred. Below this is a section titled 'My Announcements' which states 'No Institution Announcements have been posted in the last 7 days.' and 'No Course or Organisation Announcements have been posted in the last 7 days.' followed by a 'more announcements...' link.

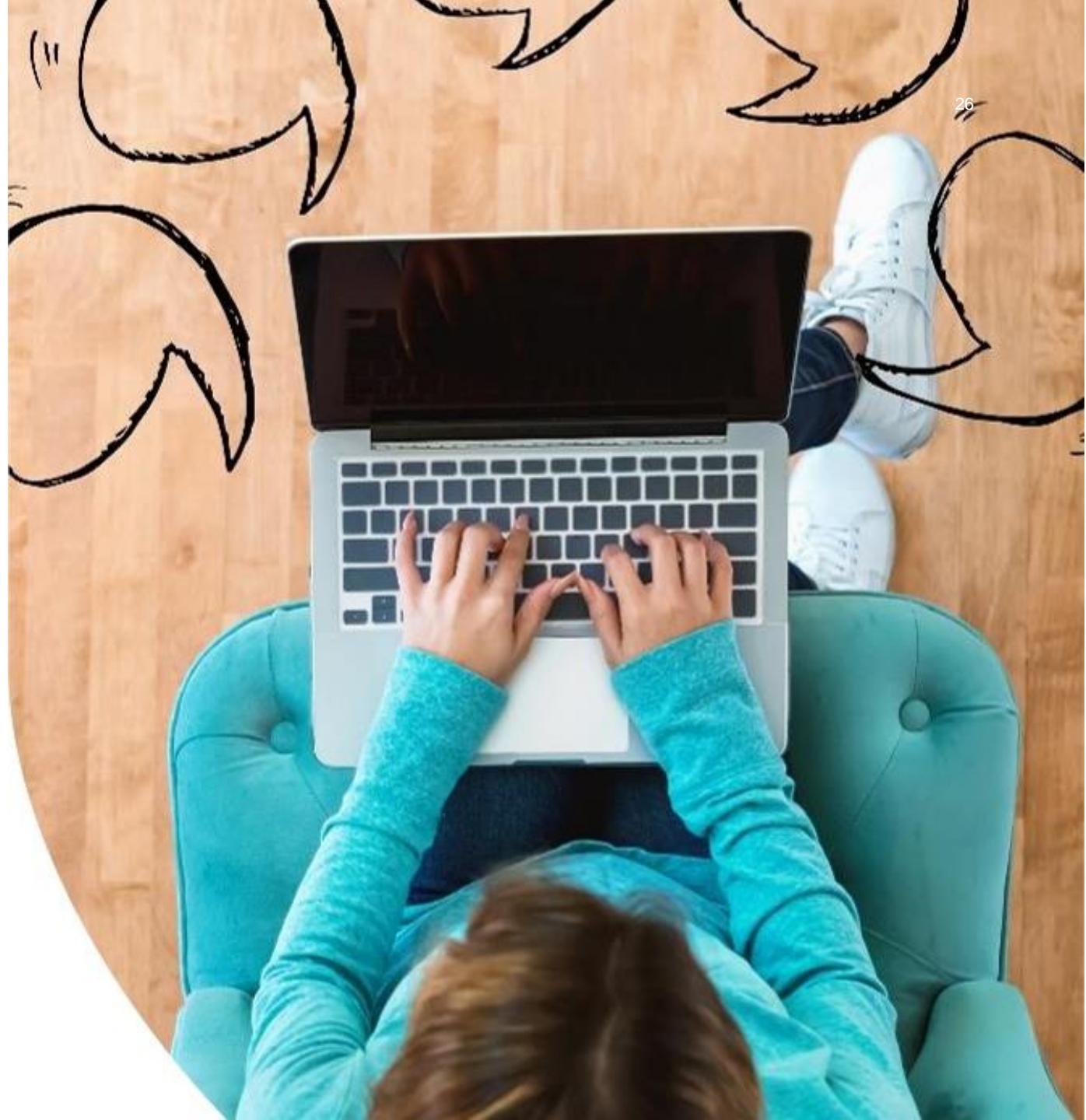
Gen AI Use in Course

- Gen AI is allowed in projects but not exam
- You will be taught how to prompt and develop larger web applications with Gen AI
- You will also learn about AI Ethics



Help and Questions

- Post general questions or issues to Ed Discussion Forum
- OR
- Email: a.bakharia1@uq.edu.au



Ed Forum

Ask any question – even anonymously (to other students but staff will know who you are)

The screenshot shows the Ed Discussion interface for the INFS3202/7202 course. The left sidebar lists courses like EECS Student Hub, INFS3202/7202, UQ CSSE Sandbox, and UQ Playground. The main area shows a thread titled "Welcome!" by Aneesha Bakharia, which reads: "Hi everyone, We're using Ed Discussion for class Q&A. This is the best place to ask questions about the course, whether curricular or administrative. You will get faster answers here from staff and peers than through email. Here are some tips: • Search before you post • Heart questions and answers you find useful • Answer questions you feel confident answering • Share interesting course related content with staff and peers All the best this semester! Aneesha (INFS3202/7202 Lecturer and Course Coordinator)".

ed THE UNIVERSITY OF QUEENSLAND AUSTRALIA INFS3202/7202 – Ed Discussion

New Thread

COURSES

- EECS Student Hub ... 1
- INFS3202/7202**
- UQ CSSE Sandbox 14
- UQ Playground 37
- Drafts 1
- Scheduled

CATEGORIES

- General
- Lectures
- Practicals/Contacts
- Weekly Quizzes
- Assignments
- Web Development S...

Search Filter

Welcome!

Aneesha Bakharia STAFF 5 days ago in General

UNPIN STAR WATCHING 110 VIEWS

Hi everyone,

9 We're using Ed Discussion for class Q&A.

This is the best place to ask questions about the course, whether curricular or administrative. You will get faster answers here from staff and peers than through email.

Here are some tips:

- Search before you post
- Heart questions and answers you find useful
- Answer questions you feel confident answering
- Share interesting course related content with staff and peers

All the best this semester!

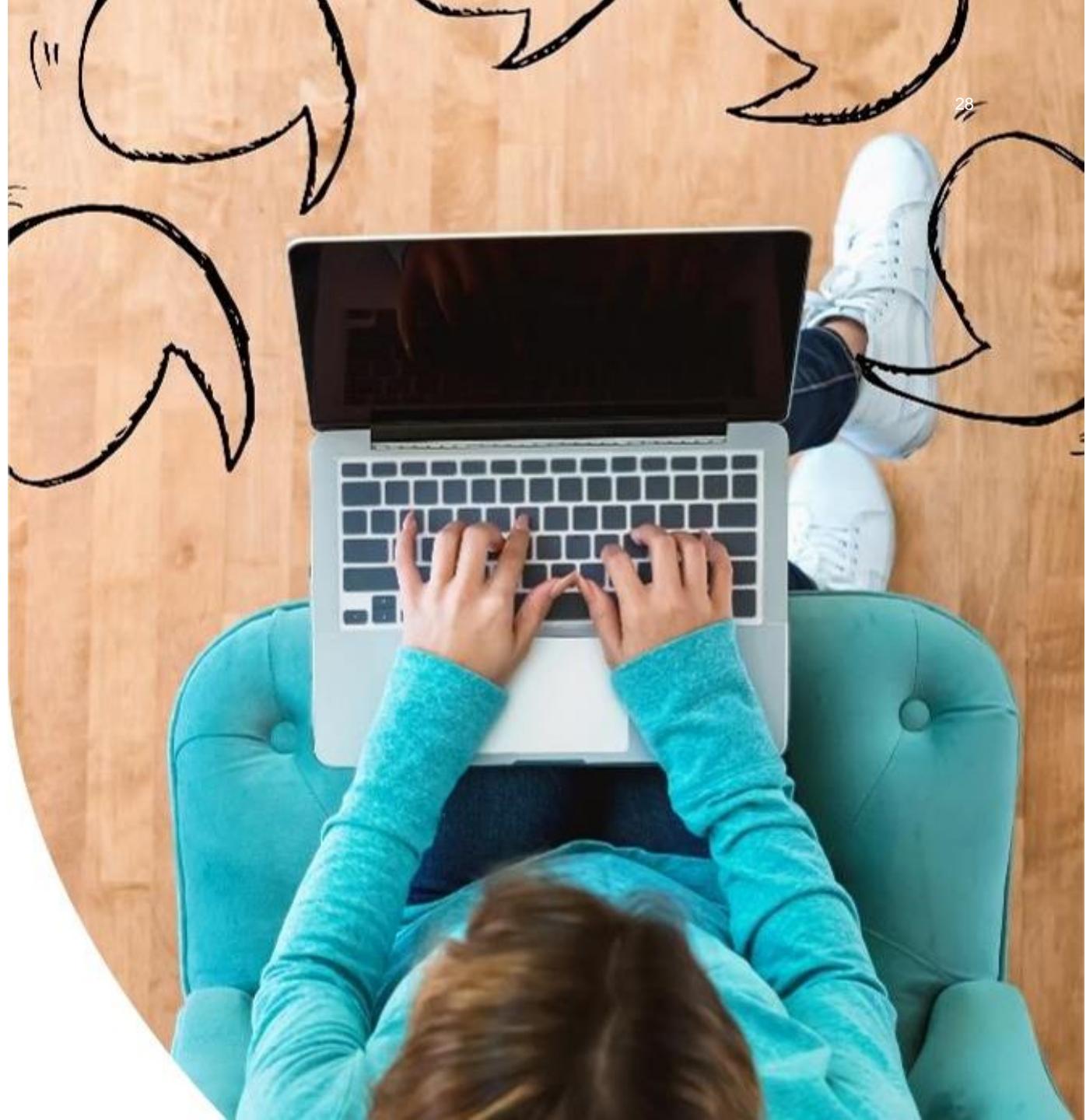
Aneesha (INFS3202/7202 Lecturer and Course Coordinator)

Add comment

- There will always be help and support on the Forum!

Feedback

- You will receive feedback for all your assessment items and in your practicals from your demonstrator
- I will sometimes send you feedback via a personalised email
 - Advice on improving
 - Additional resources
 - Congratulations for doing excellent work



Course Overview Q&A

Introduction to the World Wide Web (WWW)

[Study](#)[Research](#)[Partners and community](#)[About](#)[UQ home](#)[News](#)[Events](#)[Give](#)[Contact](#)

Search

Master the unknown

Make an impact and earn more with a UQ postgraduate degree.

[Explore more](#)

Study

Find study options to match your interests and learn how to get the most out of your time at UQ.

[Explore study options](#)

Research

Our research is creating change and making an impact on the world's biggest challenges.

[Discover our research](#)

Partners and community

We collaborate locally and globally to drive innovation.

[Get involved](#)

About UQ

We educate and work with outstanding people to build a better future.

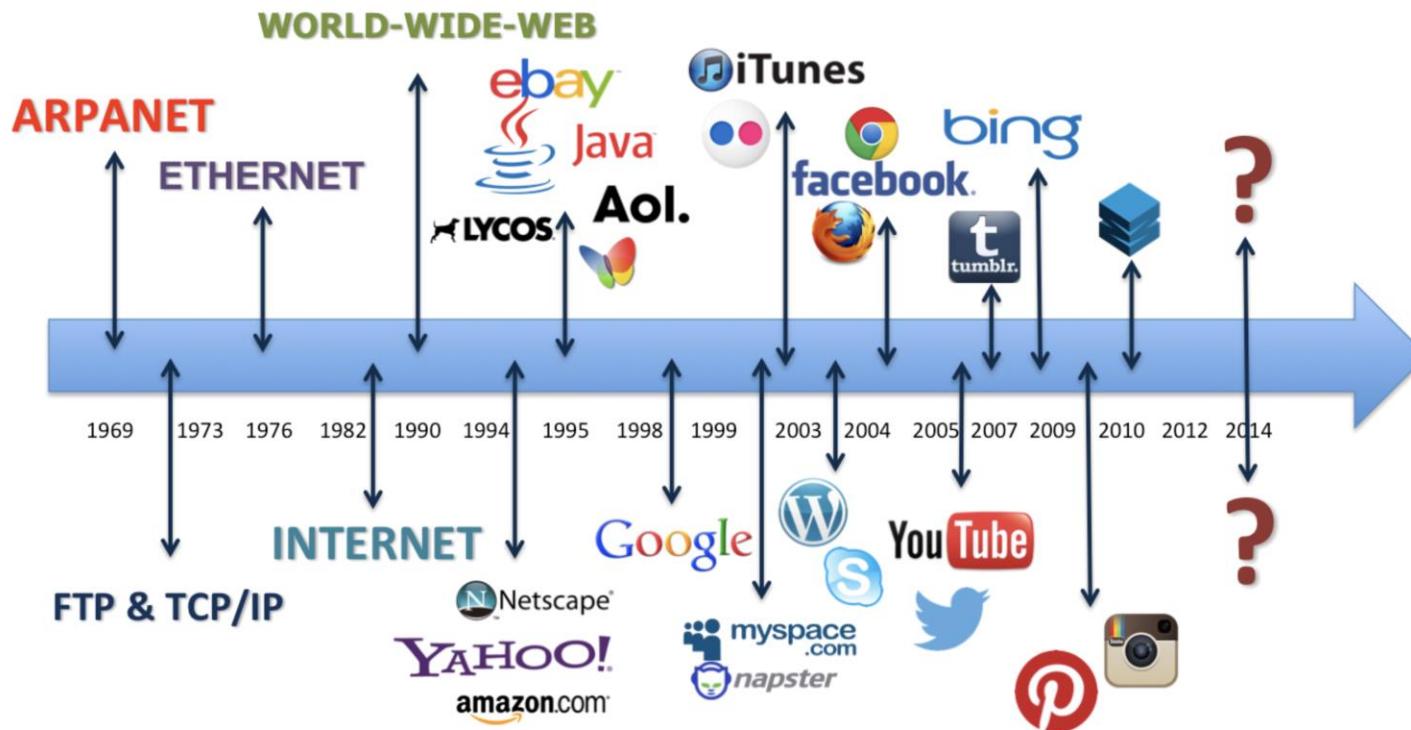
[Why choose us](#)

- You use the Web everyday
- This is the Chrome web browser displaying the UQ Website

A lot of technologies before WWW

- **Usenet (1980)** - A global discussion system allowing users to post and read messages (news or articles) on newsgroups. It was one of the earliest forms of internet communication.
- **Email (1971)** - The introduction of electronic mail by Ray Tomlinson, allowing messages to be sent between users on different hosts connected to ARPANET.
- **File Transfer Protocol (FTP, 1971)** - A standard network protocol used for the transfer of computer files between a client and server on a computer network.
- **Bulletin Board Systems (BBS, Late 1970s)** - Software running on servers connected via a dial-up modem, used for sharing messages, files, and game hosting, marking an early form of online communities.
- **TCP/IP (Transmission Control Protocol/Internet Protocol, 1983)** - The fundamental communication protocol suite for the internet was established, enabling different types of computers on various networks to communicate.
- **Domain Name System (DNS, 1984)** - A hierarchical decentralized naming system for computers, services, or other resources connected to the Internet or a private network. It translates more readily memorized domain names to the numerical IP addresses needed for locating and identifying computer services and devices.

WWW Technology Timeline



Sir Tim Berners-Lee

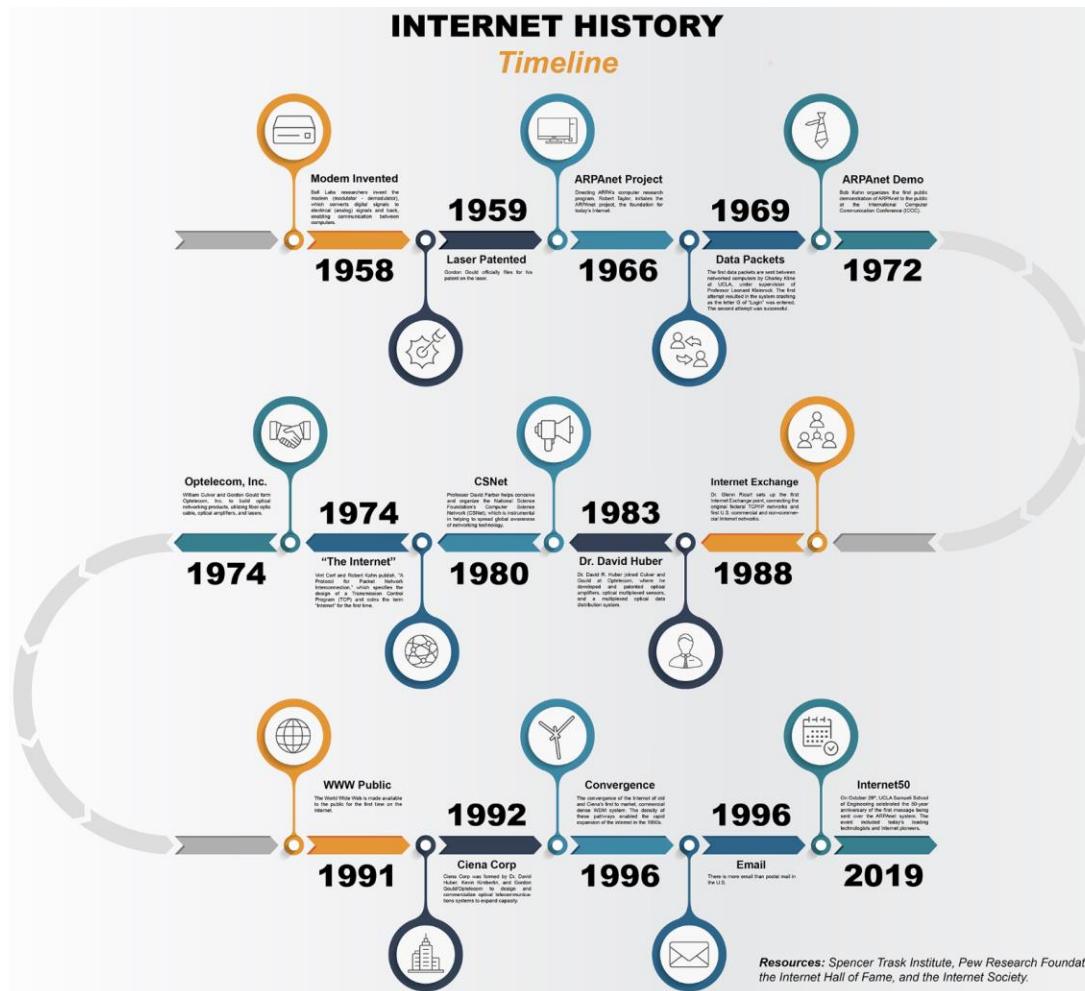
August 23, 2016 ·

...

History of website,
#internautday #internetday #internet #day
#internaut

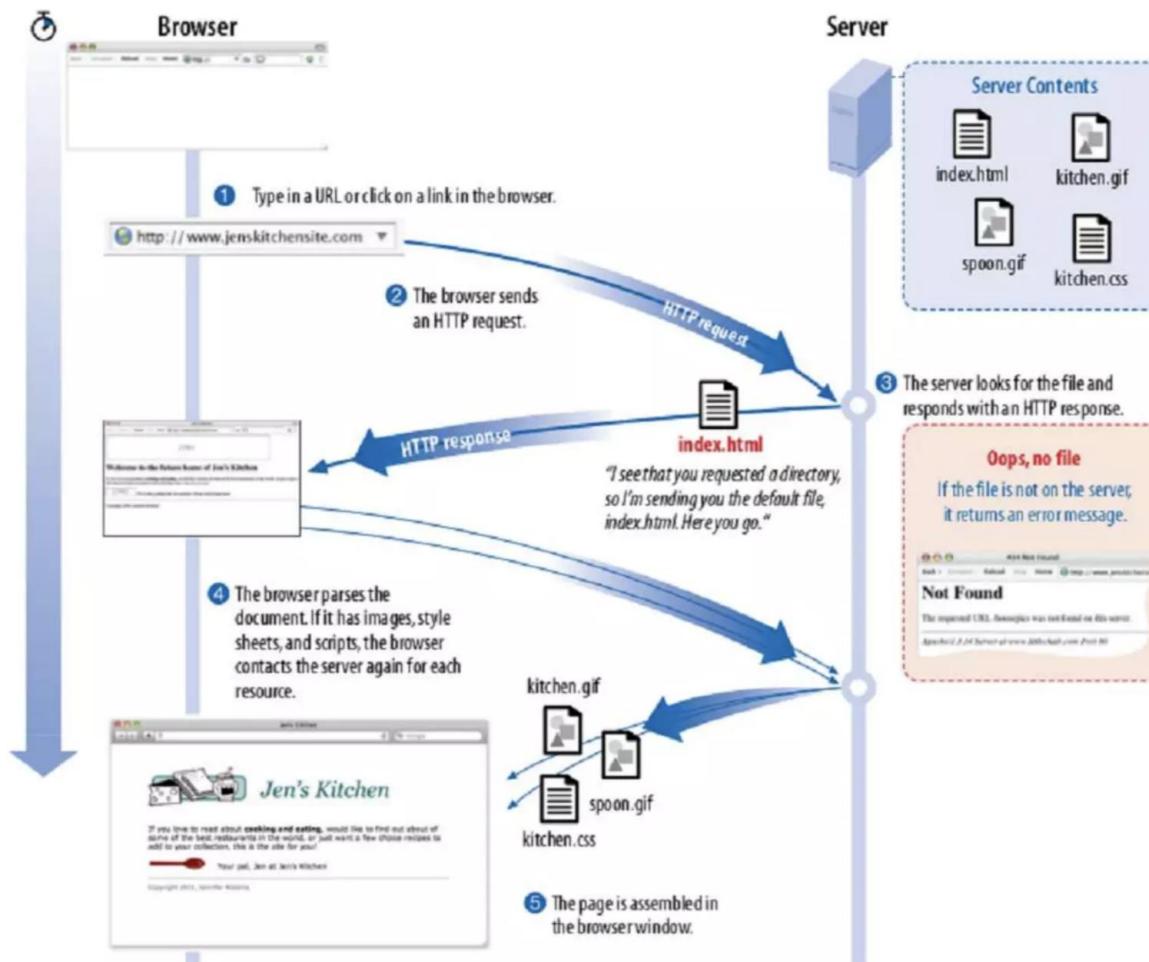
[Facebook Post](#)

WWW Technology Timeline



<https://internethistory.org/historical-timeline/>

How the Web Works?



<https://www.slideshare.net/mustafakamel/web-design-how-the-web-works>

The Invention of WWW by Sir Tim Berners Lee

Background

- Tim Berners-Lee, a British computer scientist, while working at CERN (the European Organization for Nuclear Research) in Switzerland, proposed a new way of accessing and sharing information over the internet.

The Proposal (1989)

- Berners-Lee submitted a proposal titled "Information Management: A Proposal" to management at CERN, suggesting a system to improve information flow among researchers by using hypertext to link documents.
- **Legacy and Impact**
 - Berners-Lee's invention fundamentally changed how information is shared and consumed, leading to the explosive growth of the internet and the development of the digital age.

Key Innovations of the WWW (Part 1)

- **Distributed Nature**
 - Unlike centralized networks, the WWW is fundamentally distributed across millions of servers worldwide. This means no central authority controls the information or resources, allowing for greater resilience, scalability, and freedom of access.
- **Hypertext and Hyperlinks**
 - The web introduced the use of hypertext—a method for creating links between documents or pages. This allowed users to navigate seamlessly from one document to another, creating a web of interconnected information.
- **Separate Servers for Data Hosting**
 - Information on the web is hosted on separate, independent servers rather than being stored in a single location. This means anyone can set up a web server and publish documents that are accessible to anyone in the world with an internet connection.
 - <https://www.makeuseof.com/history-of-world-wide-web/>

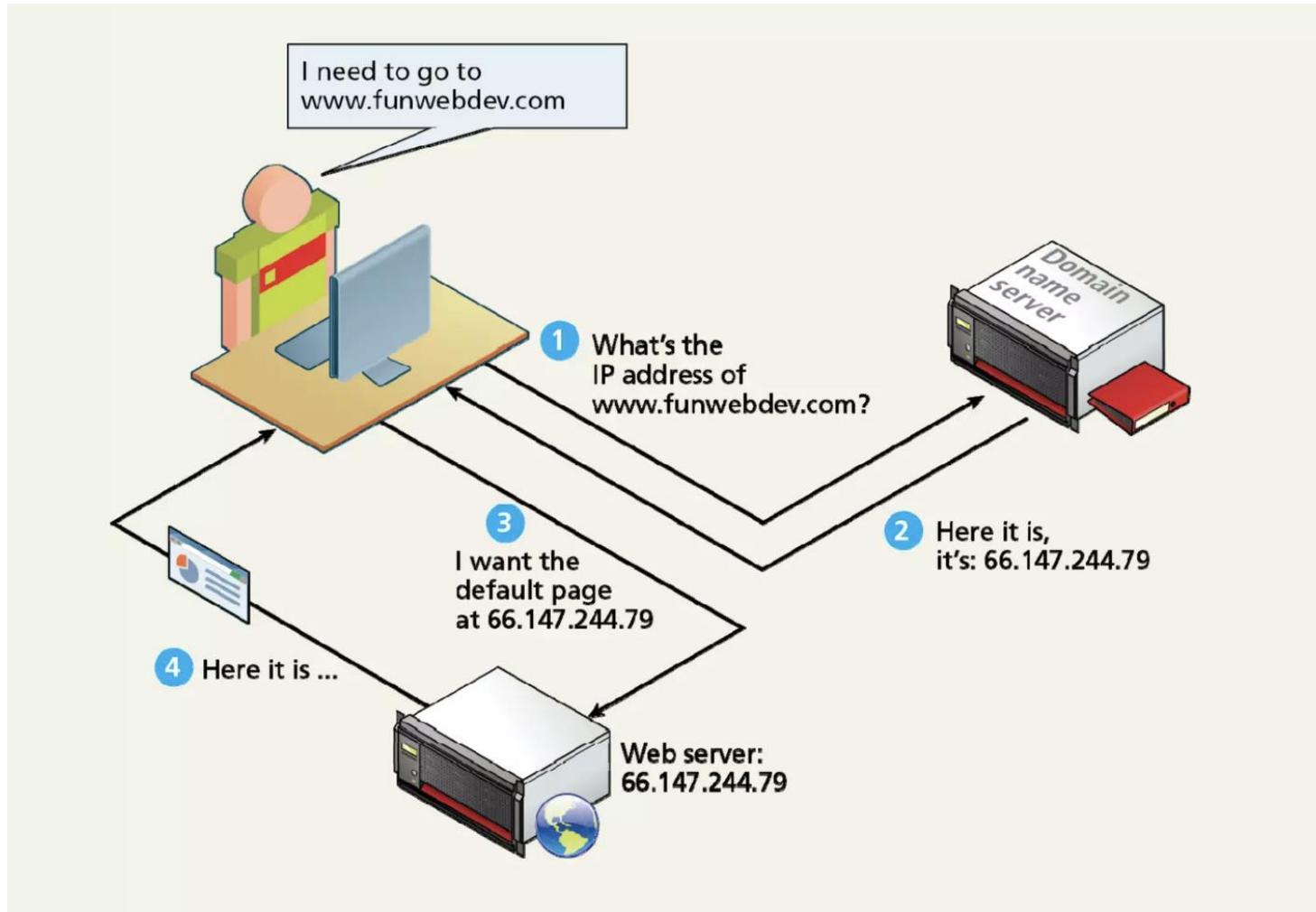
Key Innovations of the WWW (Part 2)

- **Uniform Resource Locators (URLs)**
 - URLs provide a standard way to locate resources on the web. Each web page, image, or video has a unique address that can be shared and accessed globally.
- **Platform Independence**
 - The web was designed to be accessed by any device capable of running a web browser, regardless of its operating system. This universal accessibility was a significant departure from previous systems that often required specific software or hardware.
- **Open Standards and Protocols**
 - The development of open, non-proprietary standards (such as HTML, HTTP, and later CSS and JavaScript) ensured that the web could be continuously developed and improved upon by a global community of developers and organizations.
- **User-generated Content**
 - The WWW enabled not just consumption of information but also its creation and publication by users, leading to the emergence of blogs, forums, social media, and other platforms for shared knowledge and expression.

The Domain Name Service (DNS)

- **Function:** DNS acts like the internet's phonebook. It translates human-friendly domain names (like `www.example.com`) into IP addresses (like `192.0.2.1`) that computers use to identify each other on the network.
- **Process:** When you type a website address in your browser, a DNS query is performed. This query travels through a network of DNS servers to find the IP address associated with the domain name.
- **Hierarchy:** The DNS system is hierarchical, consisting of different levels of DNS servers, including root, top-level domain (TLD), and authoritative name servers, ensuring efficient and distributed resolution of domain names to IP addresses.
- To access a website, your browser first uses DNS to resolve the site's domain name to an IP address.
- Then, it uses HTTP to send a request to the server at that IP address to fetch and display the web page.

How the Web Works?

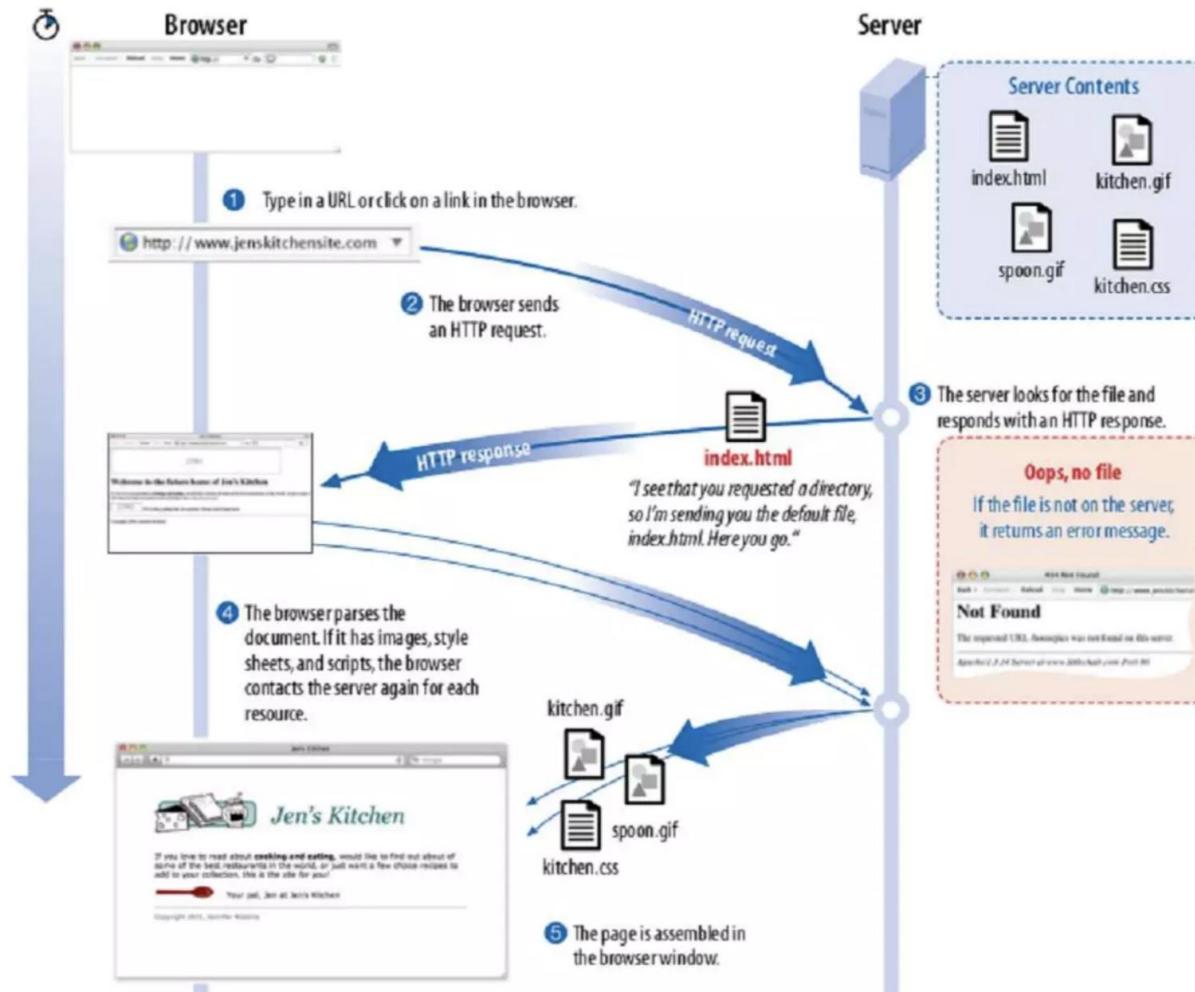


<https://www.slideshare.net/randyconnolly/chapter01-presentation-16514220>

The HyperText Transfer Protocol (HTTP/S)

- **Function:** HTTP is the protocol used for transferring web pages on the internet. It defines how messages are formatted and transmitted, and how web servers and browsers should respond to various commands.
- **Communication:** HTTP operates as a request-response protocol between a client (the web browser) and a server. The client sends an HTTP request to the server, and the server responds with an HTTP response, delivering web content.
- **Stateless Protocol:** HTTP is stateless, meaning it doesn't retain information between request-response sessions. This simplicity allows for faster communication but necessitates additional protocols (like cookies) for managing state information.
- **HTTPS:** HTTPS is the secure version of HTTP, the protocol over which data is sent between your browser and the website that you are connected to. It means all communications between your browser and the website are encrypted.

How the Web Works?



<https://www.slideshare.net/mustafakamel/web-design-how-the-web-works>

How does a Webserver Work



A web server is both hardware and software that uses HTTP (HyperText Transfer Protocol) and other protocols to respond to client requests made over the World Wide Web. The hardware aspect is the physical server that stores the web server software and the website's component files. The software aspect is the server software that understands and responds to client requests.

Key Functions of a Web Server

- **Hosting Website Files:** The web server stores the files that make up all websites, including HTML files, CSS files, JavaScript files, and images. When a user wants to visit a website, their browser sends a request to the web server hosting the site's files.
- **Processing Requests:** When the web server receives a request from a client's browser, it interprets the request, finds the requested resources (web pages, images, etc.), and sends them back to the client's browser.
- **Serving Pages:** The web server sends the requested web pages to the client's browser via HTTP. If the requested resource is not available or cannot be served for some reason, the server returns an error message (e.g., "404 Not Found").
- **Proxy for Web Frameworks:** Modern web servers can act as a reverse proxy, forwarding requests to application servers running WSGI (for Python), Node.js, or other backends. This enables dynamic content generation by integrating with databases and executing server-side logic through frameworks like Django and Next.js. This setup allows for scalable, data-driven web applications.

- Each enrolled student will get access to a UQCloud Zone
- A UQCloud Zone is a Virtual Machine (VM) that gives you access to a server with Linux, Nginx and VSCode Online.
- In Lab 1 you will learn about Nginx configuration, serving files and connecting WSGI to serve a simple Python script.

INFS3201/7202 Lab 1: UQCloud, VSCode and Python

Welcome to the first INFS3202/7202 Lab!

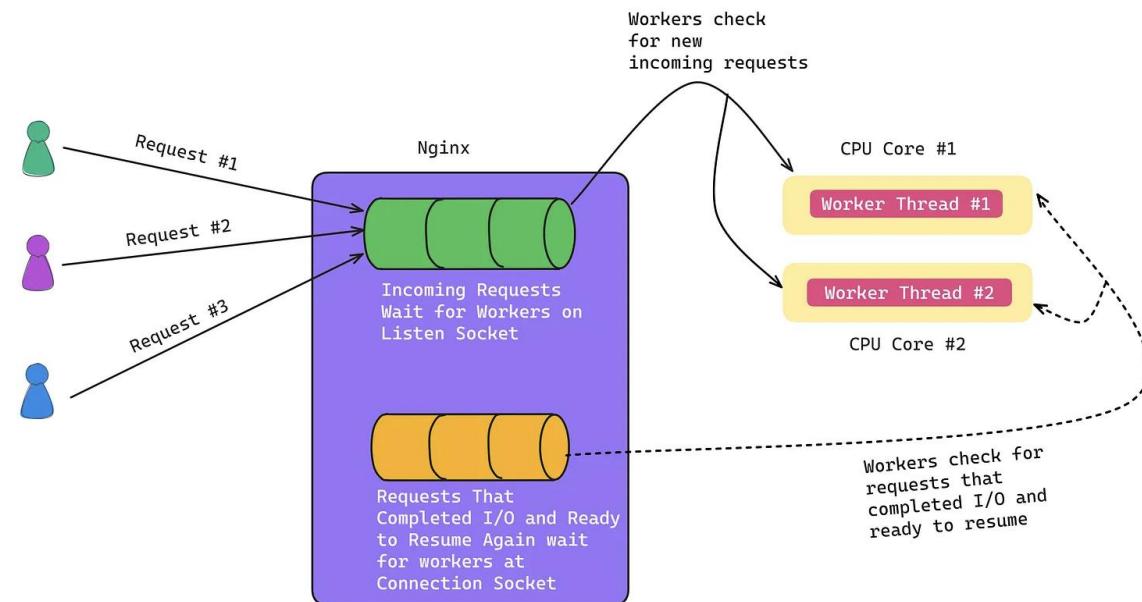
In this Lab you'll learn to:

- SSH into your UQCloud Zone and run Linux commands
- Enable VSCode Online and use it to edit a HTML page
- Enable uWSGI and run a simple Python generated page

Webservers Handle Multiple Connections

NGINX

- Nginx can handle multiple requests because it uses an asynchronous, event-driven architecture that allows it to process many connections at the same time without waiting for each one to finish.



ScalableThread.com

Fig. Nginx Approach — Event Driven

Web Browsers

A client side software to request, receive and process web pages from a Web server

- Chrome, Edge, Safari, Firefox, Chrome, Opera...
- All can handle HTML and HTTP/S
- HTML, CSS, JavaScript

Browser hosts are diversified too

- Mobile wireless devices and appliance-based



The First Web Browser



[https://en.wikipedia.org/wiki/Mosaic_\(web_browser\)#/media/File:NCSA_Mosaic_Browser_Screenshot.png](https://en.wikipedia.org/wiki/Mosaic_(web_browser)#/media/File:NCSA_Mosaic_Browser_Screenshot.png)

Example URL

```
bash
```

```
https://www.example.com:443/blog/index.html?date=20230215#section2
```

- **Scheme/Protocol:** `https`
- **Domain Name:** `www.example.com`
- **Port:** `443` (implied by HTTPS and usually not displayed)
- **Path:** `/blog/index.html`
- **Query:** `?date=20230215`
- **Fragment:** `#section2`

HyperText Markup Language (HTML)

- **Tag syntax to structure documents**
- **Definition:** The standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as CSS and JavaScript.
- **Components:** Consists of elements represented by tags (like `<html>`, `<head>`, `<body>`, `<p>`, `<div>`, etc.) that structure and define the content of web pages.
- **Role:** Serves as the skeleton of web applications, outlining the structure and content (text, images, videos).
- **Links (Hyperlinks):** Defined with the `<a>` tag, links connect one web page to another, enabling the navigational structure that is fundamental to the web's interconnectedness.
- **Forms:** Utilize the `<form>` tag along with input fields (`<input>`, `<textarea>`, etc.) to collect user data. Forms are crucial for user interactions, from search queries to login processes.

Cascading Style Sheets (CSS)

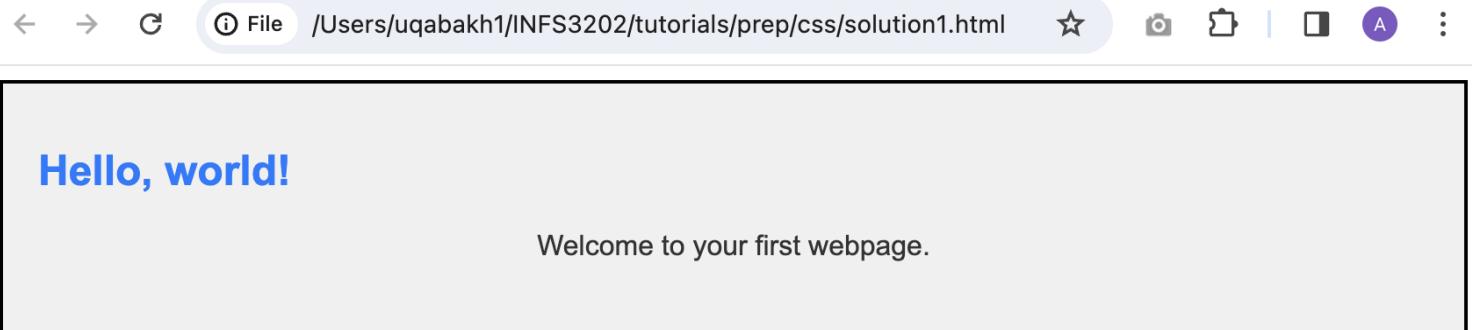
- **Definition:**
A stylesheet language used to describe the presentation of a document written in HTML. CSS describes how elements should be rendered on screen, on paper, or in other media.
- **Flexibility:**
Introduced concepts like the box model, flexbox, and grid, allowing for sophisticated layouts and responsive designs that adapt to different screen sizes.
- **Contribution:** CSS revolutionized web design by separating content from design, enabling the creation of visually engaging and user-friendly interfaces.

HyperText Markup Language (HTML)

← → ⌂ ⓘ view-source:file:///Users/uqabakh1/INFS3202/tutorials/prep/css/solution1.h... ☆

Line wrap □

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Let's Style this Web Page</title>
6     <style>
7         body {
8             font-family: Arial, sans-serif;
9             font-size: 16px;
10            color: #333;
11        }
12
13        p {
14            text-align: center;
15            margin: 20px 0;
16        }
17
18        .content-box {
19            border: 2px solid #000;
20            padding: 20px;
21            background-color: #f0f0f0;
22        }
23
24        #main-heading {
25            color: #007BFF;
26            font-size: 24px;
27        }
28    </style>
29 </head>
30 <body>
31     <div class="content-box">
32         <h1 id="main-heading">Hello, world!</h1>
33         <p>Welcome to your first webpage.</p>
34     </div>
35 </body>
36 </html>
```



Impact of HTML & CSS

- The combination of HTML's structure, the interactivity enabled by links and forms, and the sophisticated layouts made possible by CSS laid the groundwork for modern web applications.
- The separation of content (HTML), presentation (CSS), and behaviour (JavaScript) is a fundamental web development practice that ensures maintainability, accessibility, and scalability.
- Software applications that run on a web server and are accessed via a web browser over the internet. They don't require installation on the user's device.
 - But something is missing?

Introduction to Server-Side Web Programming

Server-Side Scripting and Databases

These technologies allow for the development of complex, dynamic, and responsive websites, ranging from simple informational sites to comprehensive web-based applications (like social networks, e-commerce platforms, and interactive tools).

- **Dynamic Content Generation:** Server-side scripts can generate HTML content on the fly, rather than serving static pages. This allows web applications to present customized views for each user, depending on their actions or profile data.
- **Data Processing:** These scripts are responsible for handling form submissions, processing user inputs, and performing operations like searches, data calculations, and more.
- **Interaction with Databases:** Server-side scripts can communicate with databases to store and retrieve data. This enables the functionality behind user accounts, content management systems, and other features that rely on data persistence.
- **User Authentication and Management:** Managing user sessions and authenticating users to ensure secure access to resources is a critical feature enabled by server-side scripting.
- **API Integration:** Server-side scripting often involves integrating third-party APIs to extend the functionality of web applications, such as payment gateways, social media integration, or data services.

Old Server-Side Web Development

```
<!DOCTYPE html>
<html>
<head>
    <title>PHP-HTML-JS Example</title>
    <script type="text/javascript">
        // Simple JavaScript function to toggle the table's visibility
        function toggleTable() {
            var table = document.getElementById("dataTable");
            if (table.style.display === "none") {
                table.style.display = "table";
            } else {
                table.style.display = "none";
            }
        }
    </script>
</head>
<body>

<button onclick="toggleTable()">Show/Hide Table</button>

<?php
// Database connection settings
$host = 'localhost';
$dbname = 'your_database_name';
$username = 'your_username';
$password = 'your_password';
```

```
// Create connection
$conn = new mysqli($host, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, name, value FROM example_table";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // Start of the table
    echo '<table id="dataTable" style="display: none;">';
    echo '<tr><th>ID</th><th>Name</th><th>Value</th></tr>';
    // Output data of each row
    while($row = $result->fetch_assoc()) {
        echo '<tr><td>' . $row["id"] . '</td><td>' . $row["name"] . '</td><td>' . $r
    }
    // End of the table
    echo '</table>';
} else {
    echo "0 results";
}
$conn->close();
?>

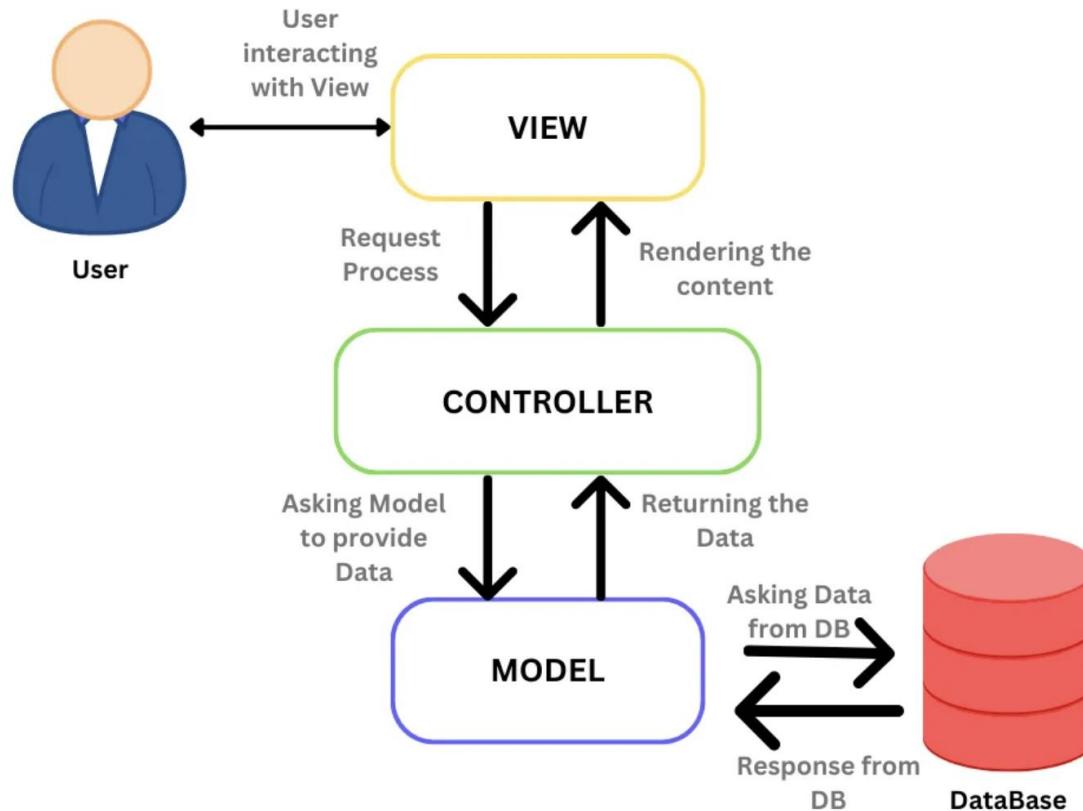
</body>
</html>
```

A lot going on in a single file

- **Database Connection:** Adjust the \$host, \$dbname, \$username, and \$password variables to match your database credentials.
- **SQL Query:** The SQL query SELECT id, name, value FROM example_table is a simple example. Customize it based on your database schema.
- **JavaScript Function:** The toggleTable() function toggles the display property of the table. It is triggered by clicking the "Show/Hide Table" button.
- **Inline PHP and HTML:** PHP is used within HTML to dynamically generate the table rows based on the database query results. Remember, mixing PHP and HTML like this is considered outdated for larger, modern applications, which often separate logic from presentation using templating engines or frameworks.

- **How can this be fixed?**

Model View Controller – Architectural Pattern



<https://medium.com/@sadikarahmantanisha/the-mvc-architecture-97d47e071eb2>

Let's convert to MVC in Django

- **Model (model.py):**

A class to describe the database. Can access the database tables using object oriented notation.

- **View (view.py):**

Fetches data from the Model and passes it to the Template.

- **Template (table.html):**

Contains the HTML and JavaScript that will be sent to the browser.

Example Django Code

Model (model.py):

```
from django.db import models

class Example(models.Model):
    id = models.AutoField(primary_key=True)
    name = models.CharField(max_length=255)
    value = models.CharField(max_length=255)

    def __str__(self):
        return self.name
```

Controller (views.py):

```
from django.shortcuts import render
from .models import Example

def example_table(request):
    data = Example.objects.all()
    return render(request, 'table.html', {'data': data})
```

Template (table.html):

```
<!DOCTYPE html>
<html>
<head>
    <title>Django Table Example</title>
    <script type="text/javascript">
        function toggleTable() {
            var table = document.getElementById("dataTable");
            if (table.style.display === "none") {
                table.style.display = "table";
            } else {
                table.style.display = "none";
            }
        }
    </script>
</head>
<body>
    <button onclick="toggleTable()">Show/Hide Table</button>

    <table id="dataTable" style="display: none;">
        <tr>
            <th>ID</th>
            <th>Name</th>
            <th>Value</th>
        </tr>
        {% for row in data %}
        <tr>
            <td>{{ row.id }}</td>
            <td>{{ row.name }}</td>
            <td>{{ row.value }}</td>
        </tr>
        {% empty %}
        <tr>
            <td colspan="3">No results found</td>
        </tr>
        {% endfor %}
    </table>
</body>
</html>
```

Advantages of MVC

- **Separation of Concerns:** MVC clearly separates the business logic (Model), user interface (View), and the user input (Controller) into different components. This separation helps manage complexity, as developers can work on individual aspects of the application without affecting the whole.
- **Development Efficiency:** By promoting a division of labor, MVC allows multiple developers to work simultaneously on the model, views, and controllers of an application, which can significantly speed up development time.
- **Maintainability:** With MVC, maintenance becomes more manageable because the modular nature of the architecture means that parts of the system can be updated or replaced with minimal impact on the rest of the application.
- **Reusability:** Components in the MVC pattern can often be reused across different parts of an application without modification, or they can be repurposed for other applications with little additional code.
- **Testability:** The decoupling of components in MVC makes it easier to write automated tests for individual parts of the codebase (unit testing), ensuring that parts of the application are working as expected independently of each other.

Week 1: Todo

- Labs start in Week 2
- Weekly RiPPLE task is due in Week 2 on Friday at 3pm
- Complete the Self-paced HTML and CSS tutorials if you need a refresh
- At the Week 2 Lecture the Design Document Assessment Item will be introduced. A Getting Started Guide will be included.

By the end of the course you should be able to go from an idea to a web application:

- Design a database
- Design and Implement the UI
- Program the functionality in a server-side programming language
- Deploy the application (in the cloud e.g. AWS)
- Use and Integrate GenAI Ethically and Creatively
 - The Web is a great way to distribute your application



CREATE CHANGE

Q&A



THE UNIVERSITY
OF QUEENSLAND
AUSTRALIA

CREATE CHANGE

Thank you