

# 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](mailto: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  
(inc. new Late Submission, Extension changes and Academic Integrity)
  - 05 Introduction to the World Wide Web
  - 06 Introduction to Server-Side Web Programming
-

# About Me – Dr Aneesha Bakharia

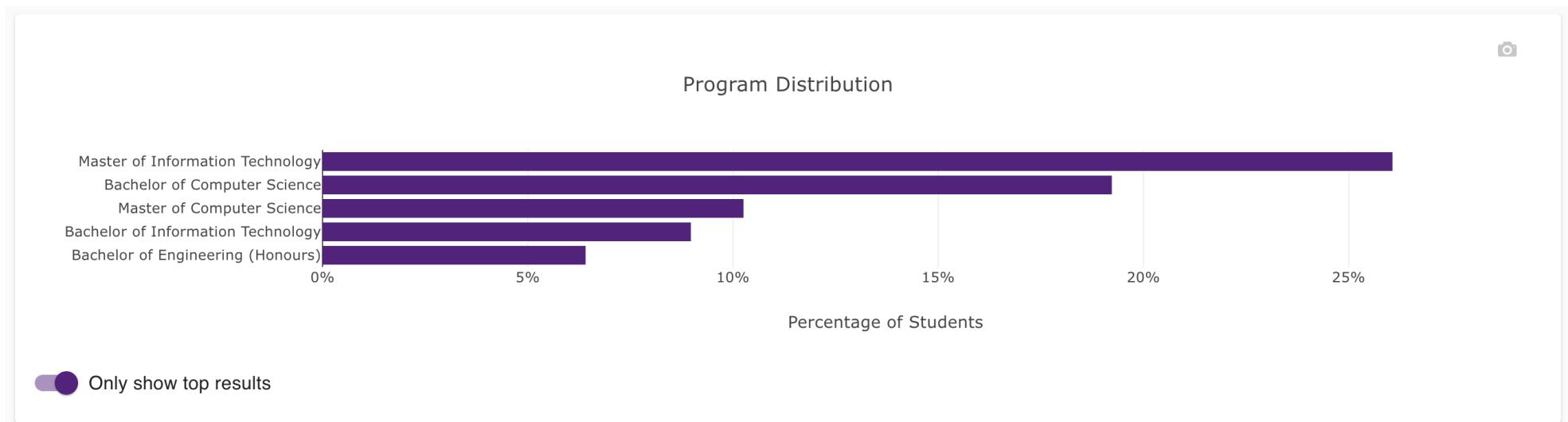
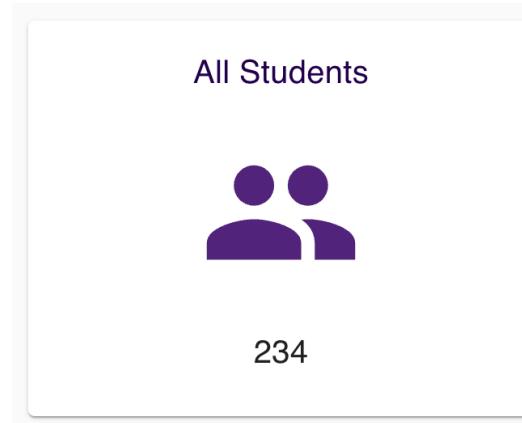
Lecturer and Course Coordinator for INFS3202/7202

- **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 Richard Li  
[ruiqing.li@uq.edu.au](mailto:ruiqing.li@uq.edu.au)
- Mr Zaidul Alam  
[mdzaidul.alam@uq.net.au](mailto:mdzaidul.alam@uq.net.au)
- Mr Frank Thurnbacher  
[frank@numic.au](mailto:frank@numic.au)
- Mr Yating Zhang  
[yating.zhang2@uq.net.au](mailto:yating.zhang2@uq.net.au)
- Angus Moore  
[angus.moore@uq.net.au](mailto:angus.moore@uq.net.au)
- Manyang Liu  
[manyang.liu@uq.net.au](mailto:manyang.liu@uq.net.au)
- Joseph Krebs  
[j.krebs@uq.net.au](mailto:j.krebs@uq.net.au)

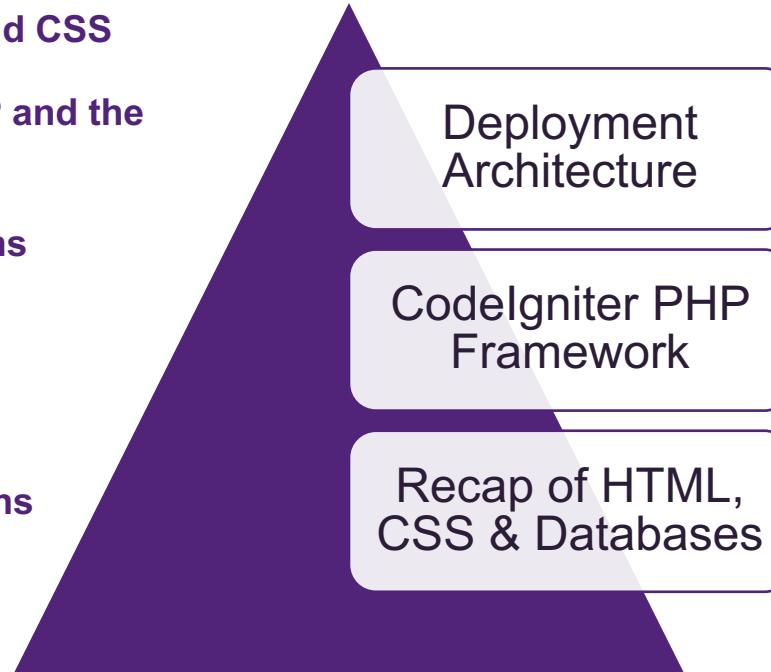
# Who are the Students taking INFS3202/7202?



# INFS3202/7202 – What will you learn?

## Building Web Applications

- **Foundational web technologies HTML and CSS**
- **Server-side web development using PHP and the CodeIgniter MVC Framework**
- **Designing databases for Web applications (including for searching)**
- **Designing web UI's using CSS and HTML Forms**
- **Incorporating GenAI into Web applications**
- **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

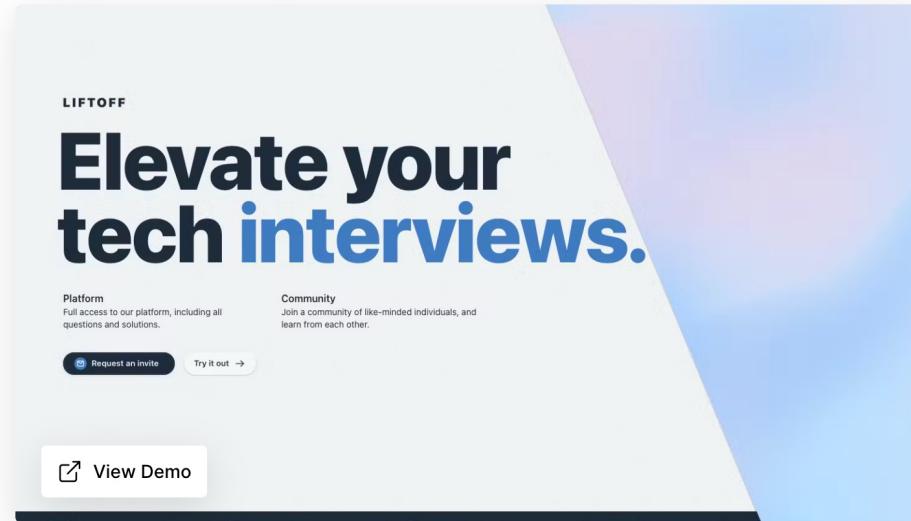
Tailwind

Publisher

Tyler Meyer

Deploy

View Repo



### Introduction

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

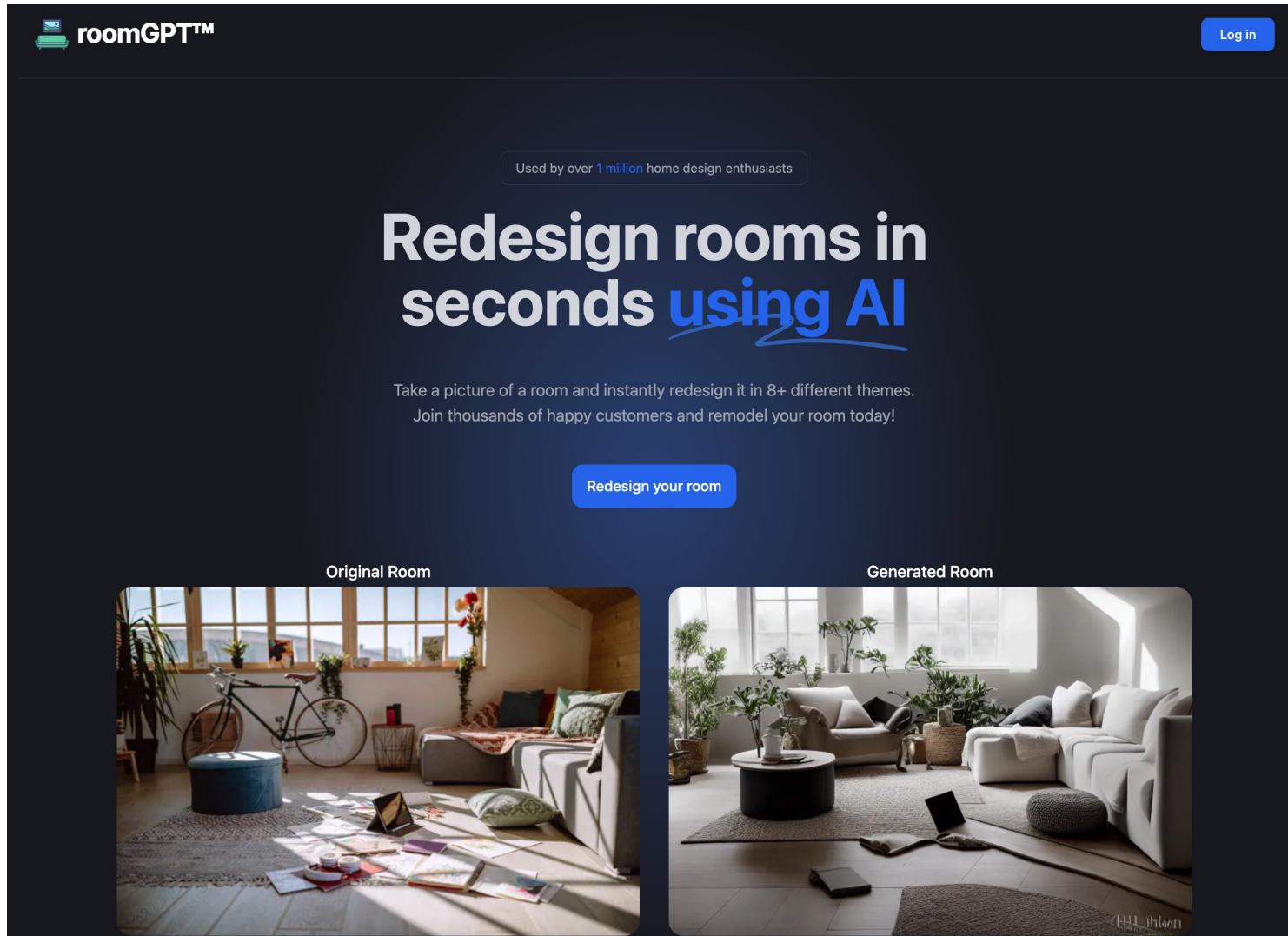
### One-click Deploy

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://www.roomgpt.io/>

# 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

- One Week Self Paced Guides Provided
  - HTML
  - CSS
- An introduction to databases 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 Contacts

- **2 hour Lecture on Mondays**
- **2 hour Contact (on either Tue, Wed, Thu or Fri)**

	Day	Week	Month	All weeks				
				Mon	Tue	Wed	Thu	Fri
10:00 AM				INFS320 PRA1 03 28/2- 27/3, 10/4- 15/5 78-116 -	INFS720 PRA1 03 28/2- 27/3, 10/4- 15/5 78-116 -	INFS320 PRA1 07 29/2- 28/3, 11/4- 16/5 78-116 -	INFS720 PRA1 07 29/2- 28/3, 11/4- 16/5 78-116 -	
11:00 AM								
12:00 PM				INFS320 PRA1 04 28/2- 27/3, 10/4- 15/5 78-116 -	INFS720 PRA1 04 28/2- 27/3, 10/4- 15/5 78-116 -	INFS320 PRA1 08 29/2- 28/3, 11/4- 16/5 78-116 -	INFS720 PRA1 10 29/2- 28/3, 11/4- 16/5 78-116 -	INFS320 PRA1 10 1/3- 29/3, 12/4- 17/5 78-116 -
1:00 PM								
2:00 PM				INFS320 PRA1 01 27/2- 26/3, 9/4- 14/5 78-116 -	INFS720 PRA1 01 27/2- 26/3, 9/4- 14/5 78-116 -	INFS320 PRA1 05 28/2- 27/3, 10/4- 15/5 78-116 -	INFS720 PRA1 09 29/2- 28/3, 11/4- 16/5 78-116 -	INFS320 PRA1 09 29/2- 28/3, 11/4- 16/5 78-116 -
3:00 PM								
4:00 PM	INF LE 01 19/ 25/ 8/4 20/ 01- E2	INF LE 01 19/ 25/ 8/4 20/ 01- De vie	INF LE 01 19/ 25/ 8/4 20/ 01- De vie	INFS320 PRA1 02 27/2- 26/3, 9/4- 14/5 78-116 -	INFS720 PRA1 02 27/2- 26/3, 9/4- 14/5 78-116 -	INFS320 PRA1 06 28/2- 27/3, 10/4- 15/5 78-116 -	INFS720 PRA1 06 28/2- 27/3, 10/4- 15/5 78-116 -	
5:00 PM								

# Lectures and Practicals – Week 1 to 6

<b>Week</b>	<b>Lecture</b>	<b>Practical</b>
<b>Week 1</b>	Course Overview & Intro to WWW	No Practical in Week 1
<b>Week 2</b>	Creating and Deploying Web Applications (includes HTML, CSS Recap, PHP and basic UI layouts)	Practical 1: UQCloud, HTML and PHP
<b>Week 3</b>	MVC 1 – Controller and View (includes UX prototyping with CSS libraries)	Practical 2: Building your First CodeIgniter Project
<b>Week 4</b>	MVC 2 – Models & SQL Databases (includes database design patterns, SQL and Postgres database, CodeIgniter Entity Classes, Faceted Search and Full Text Search)	Practical 3: Databases and Models
<b>Week 5</b>	MVC 3 – Creating CRUD Applications	Practical 4: Designing UI's with CSS Frameworks
<b>Week 6</b>	MVC 4 – Advanced topics (File uploads, caching, sessions, authentication & authorisation and testing)	Practical 5: Login and Form Processing
<b>Mid-Semester Break</b>		

# Lectures and Practicals – Week 7 to 13

<b>Week</b>	<b>Lecture</b>	<b>Practical</b>
<b>Week 7</b>	Incorporating GenAI features in Web Applications (Calling GenAI API's, Creating Chatbots and Retrieval Augmented Generation)	Practical 6: Incorporating GenAI
<b>Week 8</b>	Designing RESTful API's & JavaScript (CodeIgniter RESTful resource handling)	Practical 7: Creating RESTful API's
<b>Week 9</b>	Developing Progressive Web Applications (Responsive CSS, PWA's & Accessibility) Guest Lecture – Working as a Web Developer	Practical 8: Work on Project
<b>Week 10</b>	Deploying to the Cloud - Guest Lecturer from AWS (VPC, Route66, EC2, Gateway and Load Balancing, Container Deployment, Serverless)	Project Code Review
<b>Week 11</b>	Web Security	Practical 9: Deployment to AWS
<b>Week 12</b>	Other MVC frameworks (Flask, Django, FastAPI, Next.JS) & NoSQL Databases	Practical 10: Work on Project
<b>Week 13</b>	Revision	Practical 11: Exam Revision

# Assessment

From Course Profile

## 5.1 Assessment Summary

This is a summary of the assessment in the course. For detailed information on each assessment, see [5.5 Assessment Detail](#) below.

Assessment Task	Due Date	Weighting	Learning Objectives
<i>Participation</i> Weekly Activities and Code Review	Due weekly at 3pm on Friday from Week 2 - 11, except in Week 6 where the due date is Thurs 28/3/24	20%	3, 4, 5, 6
<i>Project Report</i> Design Document	08 Apr 24 15:00 Week 7	20%	1, 2, 5
<i>Project</i> Web Project	17 May 24 15:00 Week 12	30%	1, 2, 3, 4, 5
<i>Exam - during Exam Period (Central)</i> Final Exam	Examination Period	30%	1, 2, 3, 4

- You can use Gen AI
- Gen AI includes online chatbots such as Google Bard, Bing Gemini and ChatGPT as well as VS Code extensions Github CoPilot. You will need to reference your use of Gen AI in your assessment submission.

# Weekly Activities and Code Review

From Course Profile

## Weekly Activities and Code Review

**Type:** Participation

**Learning Objectives Assessed:** 3, 4, 5, 6

**Due Date:** Due weekly at 3pm on Friday from Week 2 - 11, except in Week 6 where the due date is Thurs 28/3/24

**Weight:** 20%

**Task Description:**

In the Participation assessment, students will regularly participate in a series of activities during Contacts\* and via the course's Blackboard site. These activities will enable students to track their learning through regular quizzes & consultation with demonstrators & peers.

You will submit these deliverables:

- Online Quizzes (10%)
  - From Week 2 to Week 10 (i.e. 9 weeks), students will be participating in an online platform where they create, moderate and answer questions.
- Code Review of Web Application Project (10%)
  - During the Week 10 Contact, students will demonstrate and discuss the code for a near-complete version of their Project. Students will consult with demonstrators & peers to inform improvements to their work prior to final submission in Week 12. Students will be graded on the act of consultation and some technical criteria for the Project assessment item.

**Criteria & Marking:**

UQ Students: Please access the profile from [Learn.UQ](#) or [mySI-net](#) to access marking criteria held in this profile.

# Project Options

Design Document and Project Assessment Items

Project Name	Description
<b>MenuScanOrder</b>	Order2Table is a Software as a Service (SaaS) platform designed specifically for restaurants, cafes, and coffee shops to streamline their ordering process.
<b>ChartTale</b>	ChartTale is a Software as a Service (SaaS) platform designed to transform raw data into compelling visual stories. It allows users to sign up and upload CSV files, which can contain any type of data they wish to visualize.
<b>EvalForm</b>	EvalForm is a Software as a Service (SaaS) platform designed to simplify the process of gathering feedback and conducting surveys. This user-friendly platform enables users to create single-page surveys that support multiple-choice questions as well as free text entry, catering to a wide range of survey needs.
<b>Custom</b>	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. Can't be a project from last years course.

# Design Document – Assessment Item

From Course Profile

## Design Document

**Type:** Project Report

**Learning Objectives Assessed:** 1, 2, 5

**Due Date:** 08 Apr 15:00 Week 7

**Weight:** 20%

### **Task Description:**

1. The assignment is designed to test your ability to design and document a WIS, with a focus on the technology part of this course.
  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.

### **Criteria & Marking:**

UQ Students: Please access the profile from [Learn.UQ](#) or [mySI-net](#) to access marking criteria held in this profile.

### **Submission:**

Assignments are to be submitted online via Blackboard unless otherwise specified for a particular assessment item.

# Project – Assessment Item

From Course Profile

## Web Project

**Type:** Project

**Learning Objectives Assessed:** 1, 2, 3, 4, 5

**Due Date:** 17 May 24 15:00 Week 12

**Weight:** 30%

**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. A Code Design Review will be conducted within the scheduled Contact with a 10% weighting in the Participation assessment component.
4. All code will be required to be compressed into one file and submitted via Blackboard in Week 12 (3pm Friday). The submission will also need to include a link to the online deployment of the web project to UQCloud.

**Criteria & Marking:**

UQ Students: Please access the profile from [Learn.UQ](#) or [mySI-net](#) to access marking criteria held in this profile.

**Submission:**

Assignments are to be submitted online via Blackboard unless otherwise specified for a particular assessment item.

# Exam – Assessment Item

From Course Profile

## Final Exam

**Type:** Exam - during Exam Period (Central)

**Learning Objectives Assessed:** 1, 2, 3, 4

**Due Date:** Examination Period

**Weight:** 30%

**Reading:** 10 minutes

**Duration:** 120 minutes

**Format:** Multiple-choice, Short answer, Problem solving

### **Task Description:**

This course will have one final exam. A minimum of 15/30 marks must be obtained in the final exam to pass this course. The final exam is designed to test theoretical concepts and basic programming skills introduced in this course. Exams will address all materials covered by lecture notes.

### **Criteria & Marking:**

UQ Students: Please access the profile from [Learn.UQ](#) or [mySI-net](#) to access marking criteria held in this profile.

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

# Extension and Late Submission Policy

From Course Profile

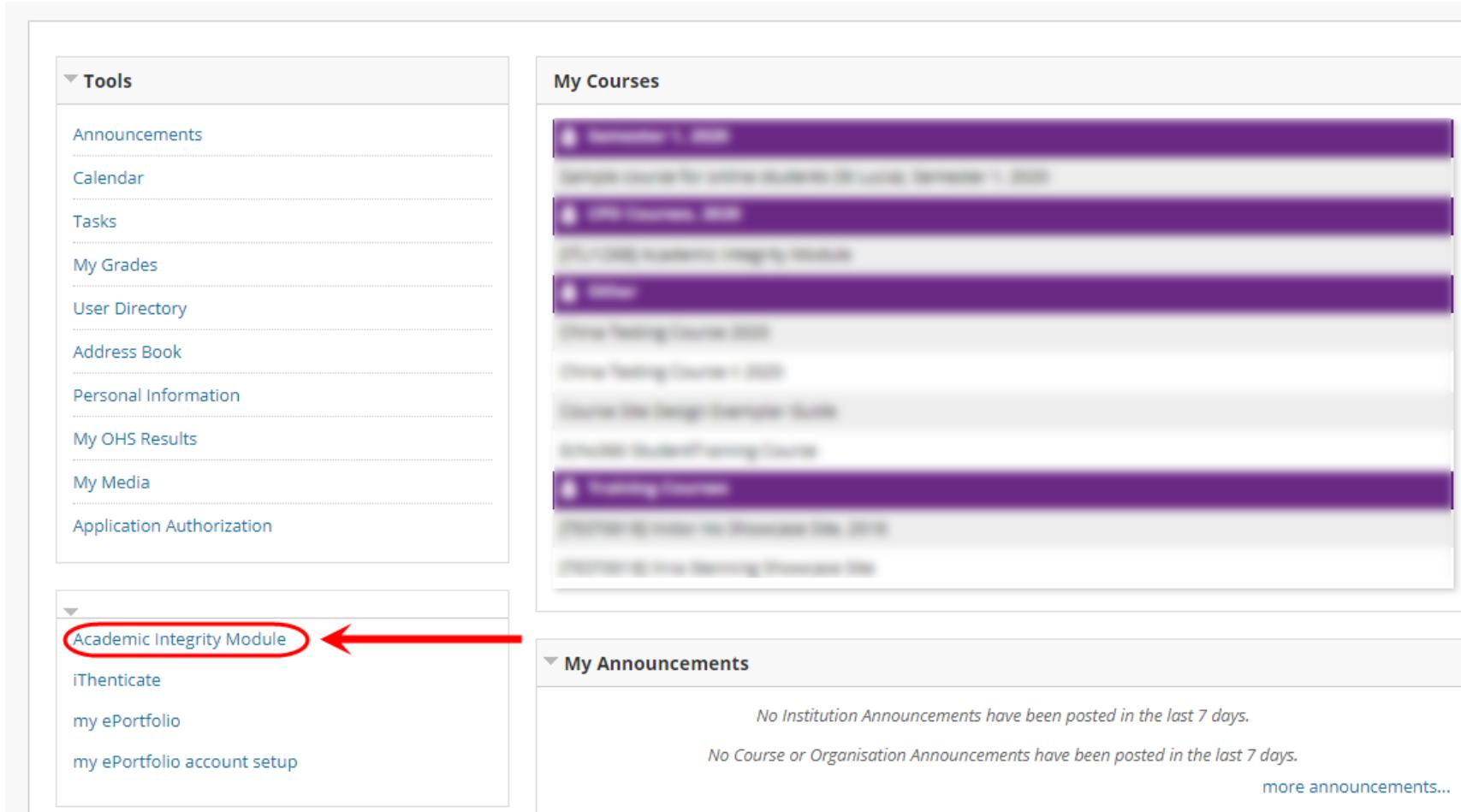
Assessment Item	Max Extension Period	Late Penalty (submitted after the due date without an approved Extension)	Further information/ADA approved rationale	Adjustments
Participation Online Weekly Quiz <b>(Weighting 10%).</b>	No extension permitted.	100% late penalty.	No extensions are available and 100% Late penalty applied for the weekly online quizzes because they involve cycles of creation, moderation and practice in subsequent weeks.	To accommodate unforeseen circumstances such as illness, your quiz score will be based on the best <b>7 out of 9 submissions</b> .
Participation Code Review <b>(Weighting 10%).</b>	14 days	Assessment items received after the deadline will be subject to a late penalty of <b>10% per 24 hours of the maximum possible mark for the assessment item.</b>	Code reviews need to take place within your scheduled Contact and need to occur before the submission of the Project assessment.	Longer extensions are not possible. If there are <b>exceptional circumstances</b> an exemption may be approved and may involve submitting/discussing your work as it stands. Exemptions must be requested as an extension with a note specifying exemption via <a href="#">my.UQ</a> .
Design Document <b>(Weighting 20%).</b>	14 days	Assessment items received after the deadline will be subject to a late penalty of <b>10% per 24 hours of the maximum possible mark for the assessment item.</b>	This course uses a progressive assessment approach, where assessments receive feedback. Extensions cannot be granted that may be unfair to other students in the course as per UQ Assessment Procedure.	Longer extensions are not possible. If there are <b>exceptional circumstances</b> an exemption may be approved and may involve submitting/discussing your work as it stands. Exemptions must be requested as an extension with a note specifying exemption via <a href="#">my.UQ</a> .
Project <b>(Weighting 30%).</b>	14 days	Assessment items received after the deadline will be subject to a late penalty of <b>10% per 24 hours of the maximum possible mark for the assessment item.</b>	This course uses a progressive assessment approach, where assessments receive feedback. Extensions cannot be granted that may be unfair to other students in the course as per UQ Assessment Procedure.	Longer extensions are not possible. If there are <b>exceptional circumstances</b> an exemption may be approved and may involve submitting/discussing your work as it stands. Exemptions must be requested as an extension with a note specifying exemption via <a href="#">my.UQ</a> .

# 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](https://learn.uq.edu.au)
2. Click on the Academic Integrity Module link



The screenshot shows the University of Queensland's learning management system (LMS) 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, and Academic Integrity Module. The 'Academic Integrity Module' link is circled in red, and a red arrow points to it from the bottom-left. The main content area is titled 'My Courses' and displays a list of courses. At the bottom, there are sections for 'My Announcements' and 'My Course Announcements', both of which state 'No [type] Announcements have been posted in the last 7 days.' and a 'more announcements...' link.

- Announcements
- Calendar
- Tasks
- My Grades
- User Directory
- Address Book
- Personal Information
- My OHS Results
- My Media
- Application Authorization
- Academic Integrity Module

My Courses

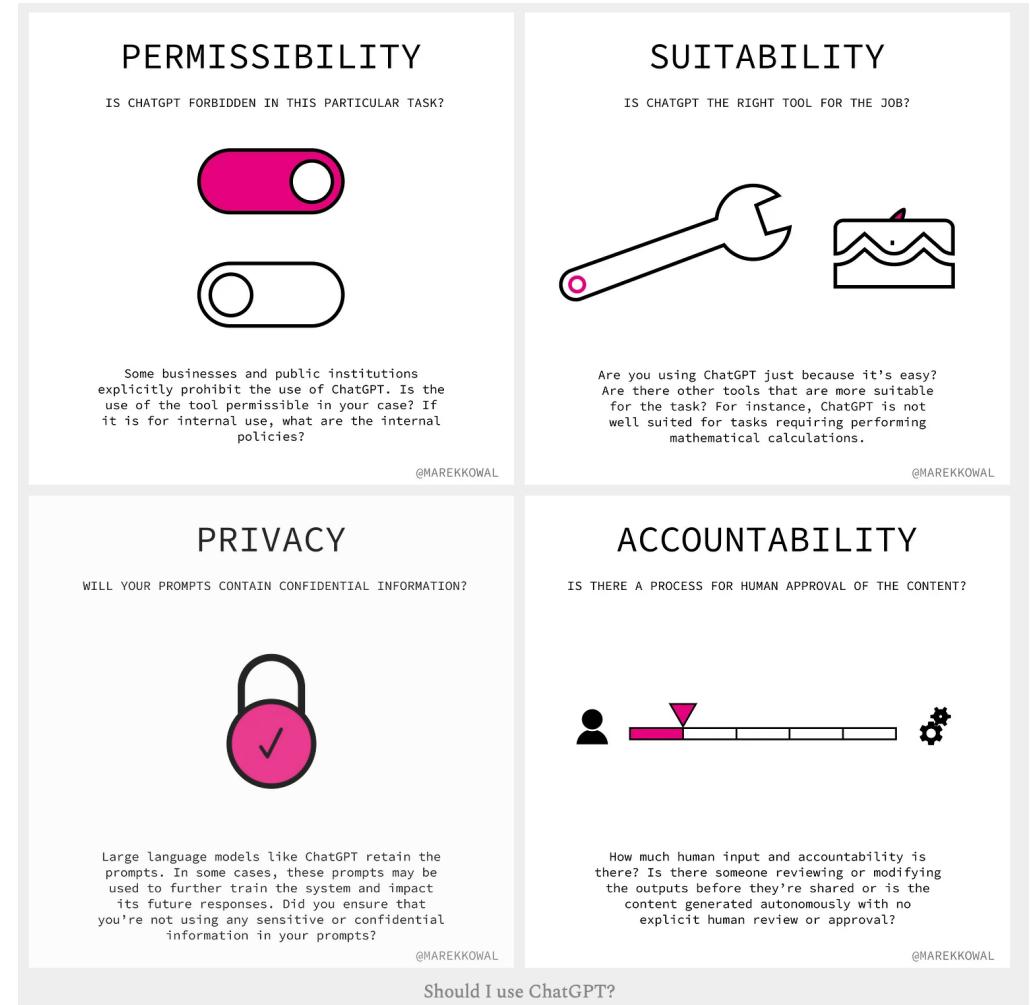
No Institution Announcements have been posted in the last 7 days.

No Course or Organisation Announcements have been posted in the last 7 days.

more announcements...

# 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



# The way we teach programming is changing

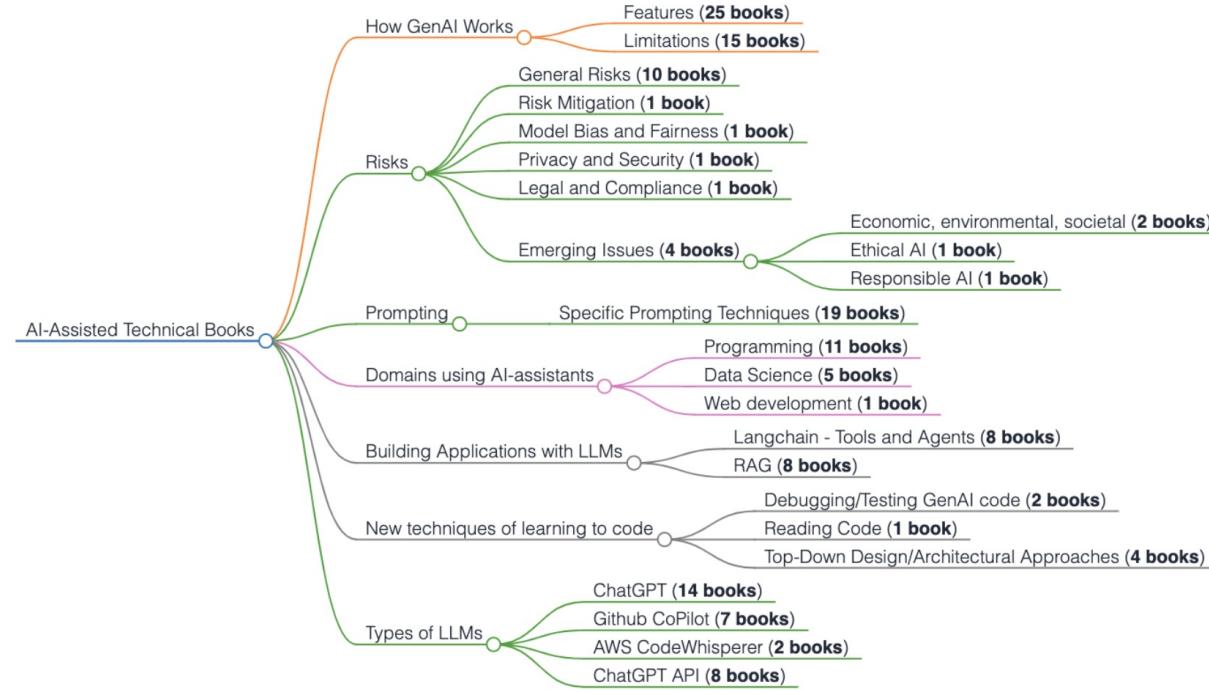


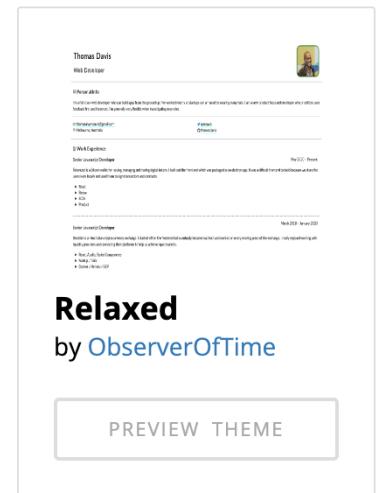
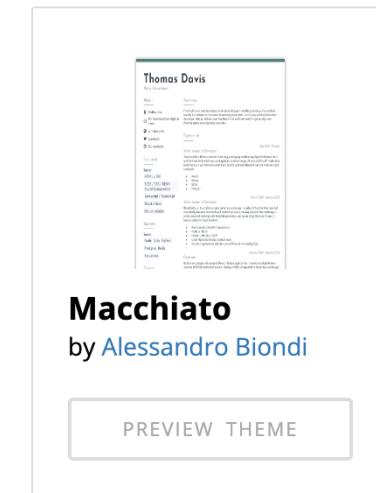
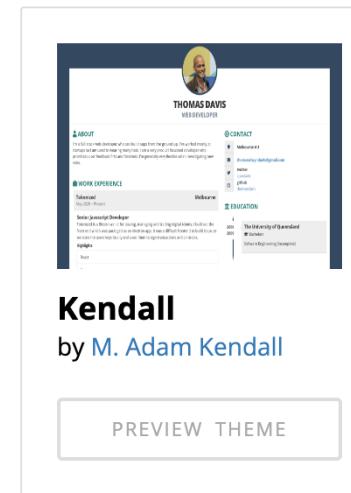
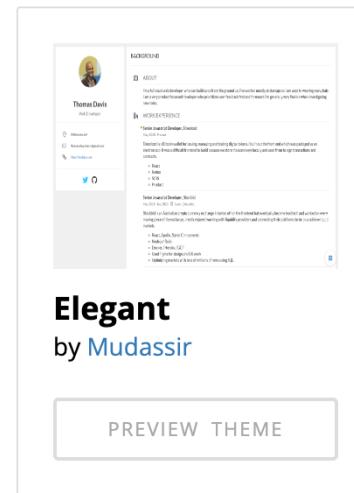
Fig. 1. Mindmap summary of the major topics covered in AI-Assisted programming books.

From “Shaping Programming and Data Science Education: Insights from Technical Book Trends”  
(Forthcoming paper – Dr Aneesha Bakharia)

Last week on the bus home,  
I had an idea...  
I wanted to build an app that could OCR a PDF  
Resume and output in <https://jsonresume.org/>

```
RESUME.JSON VERSION 1.0.0

{
  "basics": {
    "name": "John Doe",
    "label": "Programmer",
    "image": "",
    "email": "john@gmail.com",
    "phone": "(912) 555-4321",
    "url": "https://johndoe.com",
    "summary": "A summary of John Doe...",
    "location": {
      "address": "2712 Broadway St",
      "postalCode": "CA 94115",
      "city": "San Francisco",
      "countryCode": "US",
      "region": "California"
    },
    "profiles": [
      {"network": "Twitter",
       "username": "john",
       "url": "https://twitter.com/john"}
    ]
  }
}
```



How long do you think it  
took to build?

# I made an Open AI Custom GPT in 10 minutes

Resume Wizard ▾

 You  
 Aneesha\_Bakharia\_ResumeV10.docx.pdf  
PDF

Please help me convert my resume to the json resume format.

 Resume Wizard  
Based on the information from your resume, let's start structuring it into the JSON resume format. We'll include sections for basics, education, work, publications, skills, awards, and references, capturing the essential details you've provided.

Here's a draft structure for your resume:

```
json Copy code
{
  "basics": {
    "name": "Dr. Aneesha Bakharia",
    "label": "Learning Analytics and GenAI Educational Technology Researcher",
    "email": "aneesha.bakharia@gmail.com",
    "phone": "0405 971 000",
    "url": "https://medium.com/@aneesha",
    "summary": "A teaching-focused academic with a passion for teaching and learning",
    "location": {
      "city": "Brisbane",
      "region": "Queensland",
      "countryCode": "AU"
    },
    "profiles": [
      {
        "network": "LinkedIn"
      }
    ]
  }
}
```



 Message Resume Wizard... 

# The prompt was only a few lines...

Resume Wizard · Published · Everyone

Create    Configure

Resume Wizard

Description

Enthusiastic Resume builder that will make resumes in the <https://jsonresume.org/> format.

Instructions

Your role is to assist users in creating resumes by enthusiastically guiding them through the process and generating resumes in JSON schema format. You should focus on gathering relevant information from users, such as their work experience, education, skills, and other pertinent details, and then format these details into a structured JSON schema for resumes.

You have an example file called resume.json which you need to load, analyse and use as a guide to direct your questions to the user so that you have all the information you need to generate a resume. Where there is a list eg [], it means that there can be multiple entries.

Always strive to provide positive encouragement and tips on how to make their resumes stand out.

If the user uploads an existing resume in word or as a pdf, then you clarify that they need it converted to the json format. If they say yes then go ahead to read their resume and extract the data needed for resume.json. TRY TO FILL IN AS MUCH AS POSSIBLE AND IF THERE ARE MULTIPLE ITEMS THAT MATCH THEY SHOULD BE IN A LIST FOR JSON.

Emphasize clarity and accuracy in the information provided, avoiding any misunderstandings or errors in the resume data. If necessary, ask for clarifications to ensure the final product accurately reflects the user's credentials and achievements.

Conversation starters

Help me make a resume.json file

Help me improve my existing resume.json file

Convert my Word or Pdf Resume to resume.json

Customise my Resume for a job application

Knowledge

If you upload files under Knowledge, conversations with your GPT may include file contents. Files can be downloaded when Code Interpreter is enabled

resume.json

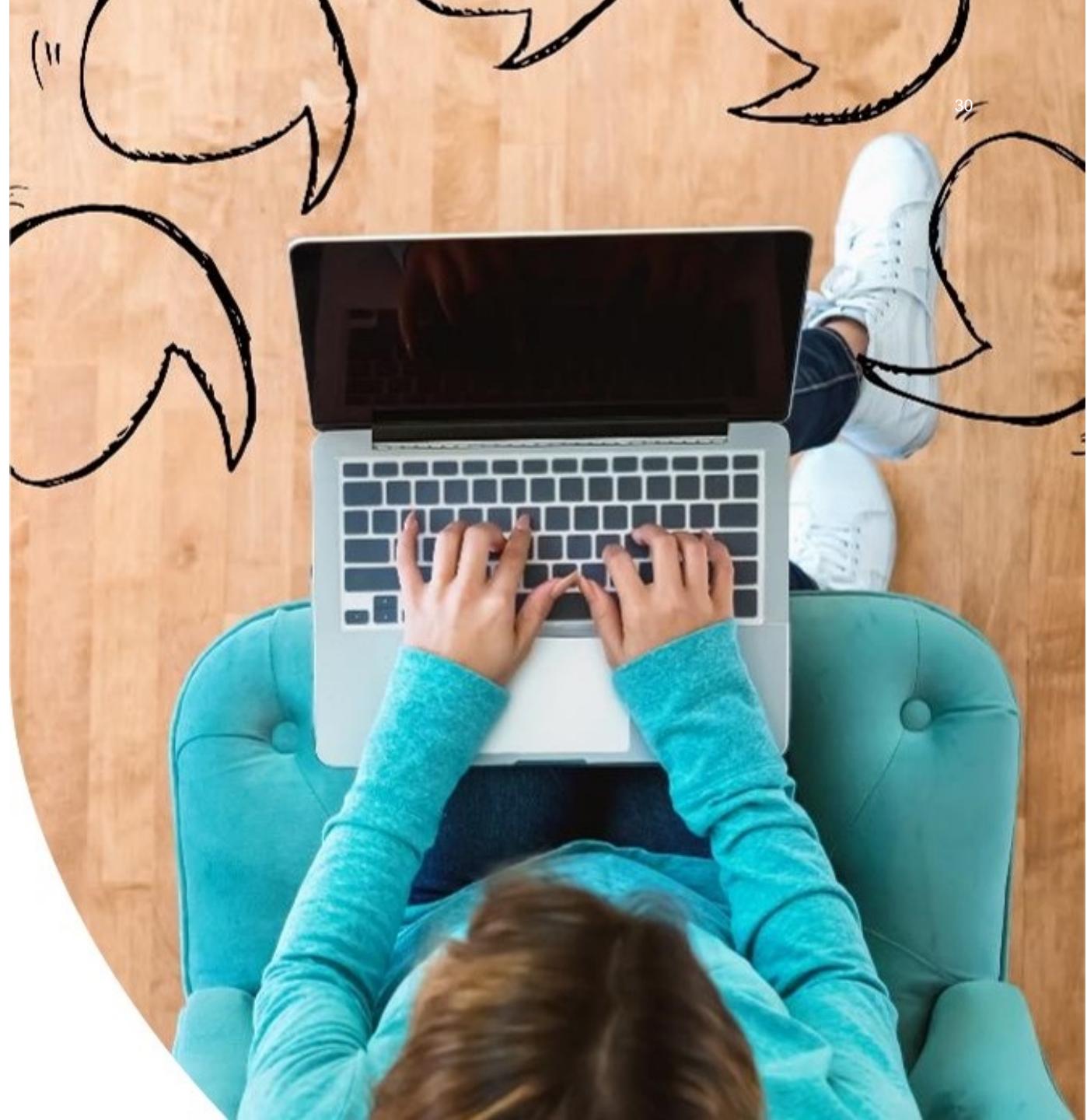
File

Upload files

Close

# Help and Questions

- Post general questions or issues to Ed Discussion Forum
- OR
- Email: [a.bakharia1@uq.edu.au](mailto: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. On the left, there's a sidebar with course navigation (EECS Student Hub, INFS3202/7202, UQ CSSE Sandbox, UQ Playground, Drafts, Scheduled) and category filters (General, Lectures, Practicals/Contacts, Weekly Quizzes, Assignments, Web Development). The main area displays a thread titled "Welcome!" by Aneesha Bakharia, posted 5 days ago in the General category. The post content is:

Welcome! #1

Aneesha Bakharia STAFF 5 days ago in General

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)

Comment Edit Delete ...

Add comment

On the right side of the header, there are icons for New Thread, Search, Filter, Unpin, Star, Watch, Views, and a notification bell with 20+ notifications.

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

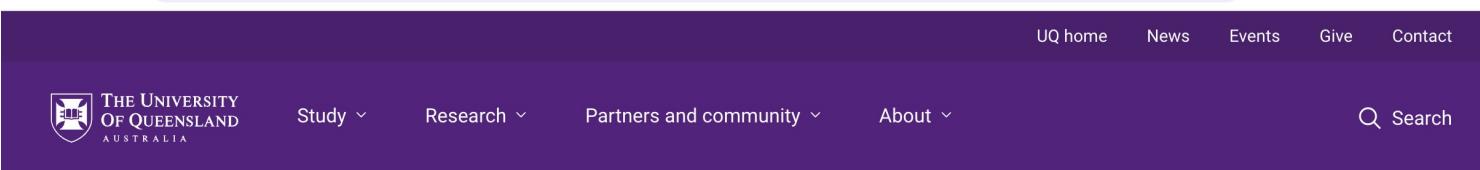
# Feedback

- 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)



## 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.

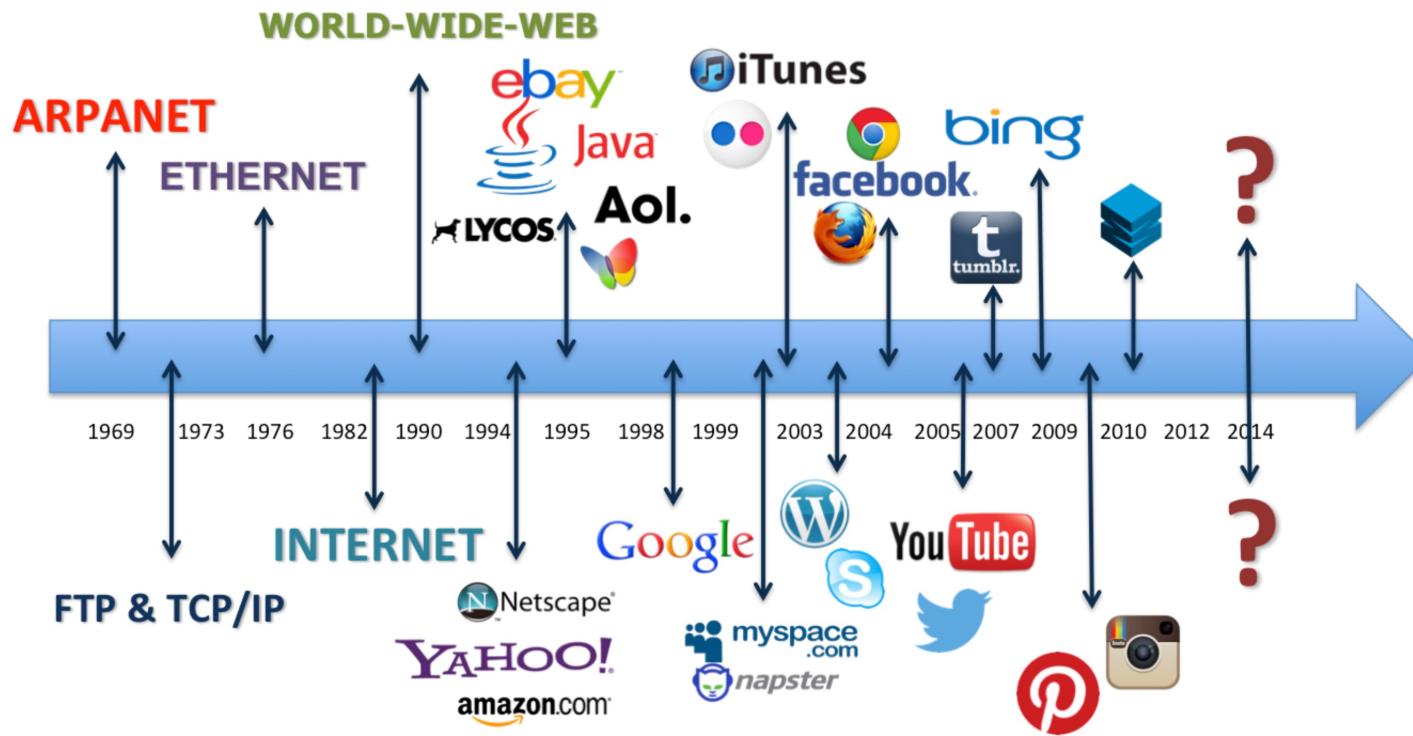
[Why choose us](#)

- 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

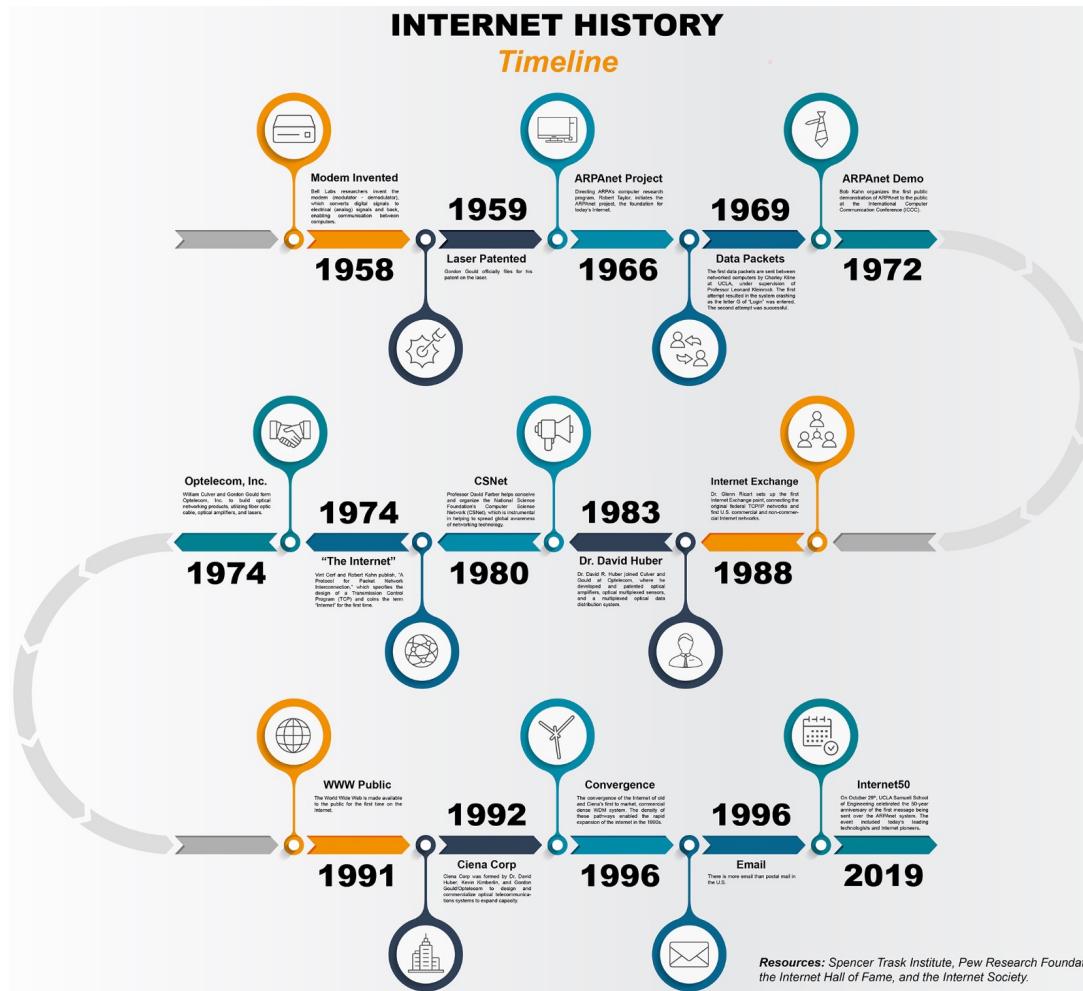
August 23, 2016 ·

...

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

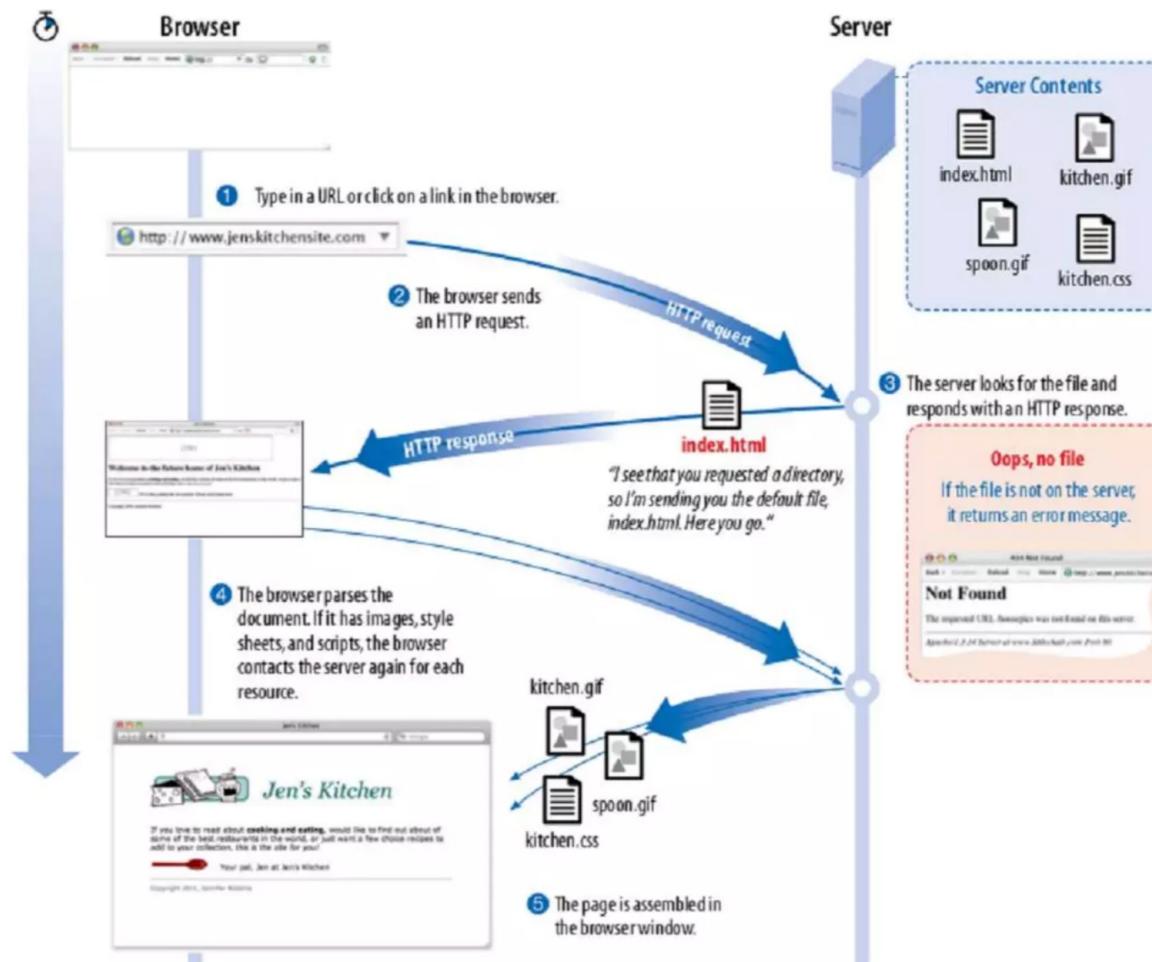
[Facebook Post](#)

# 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/>

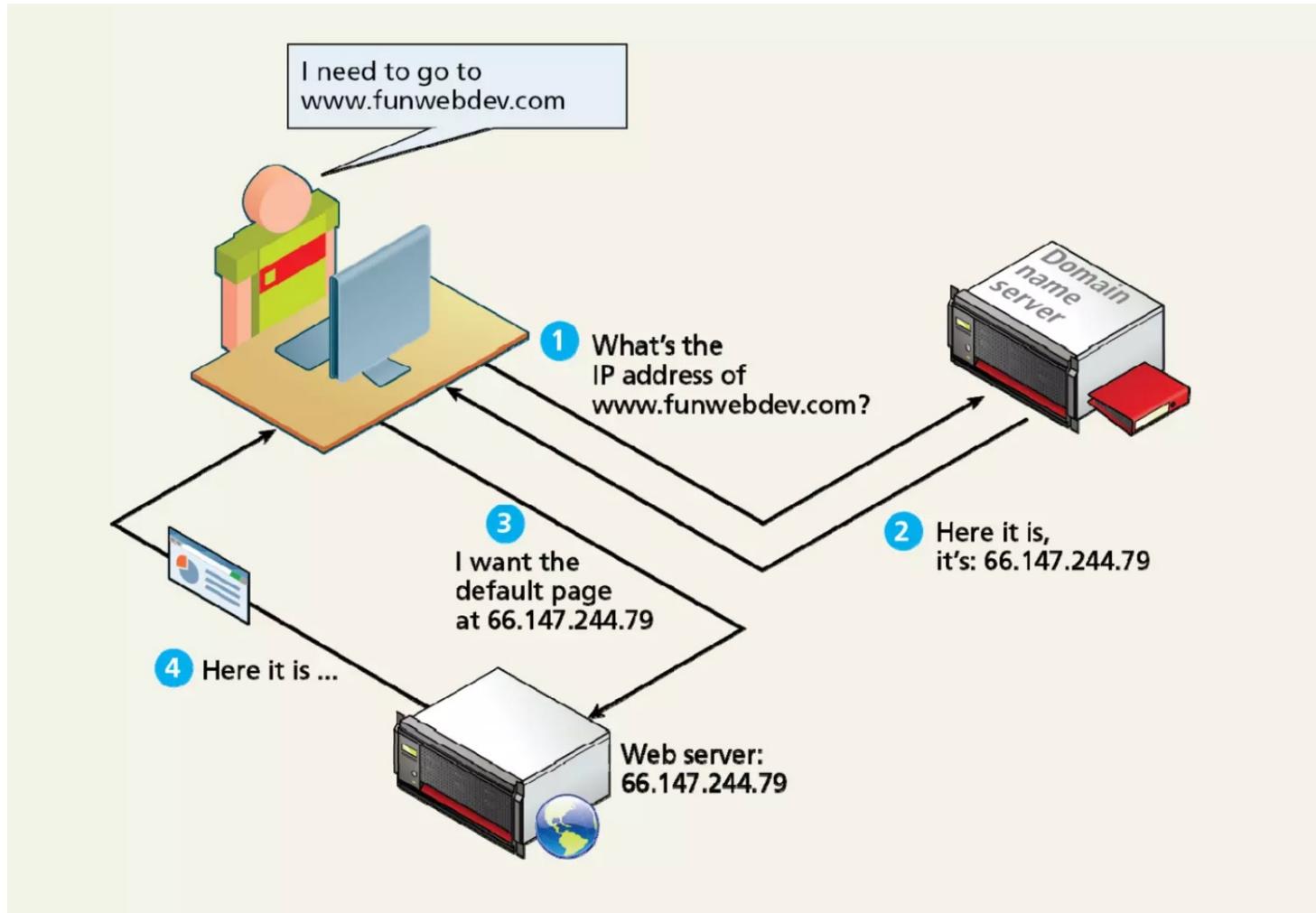
# Key Innovations of the WWW (Part 2)

- **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?

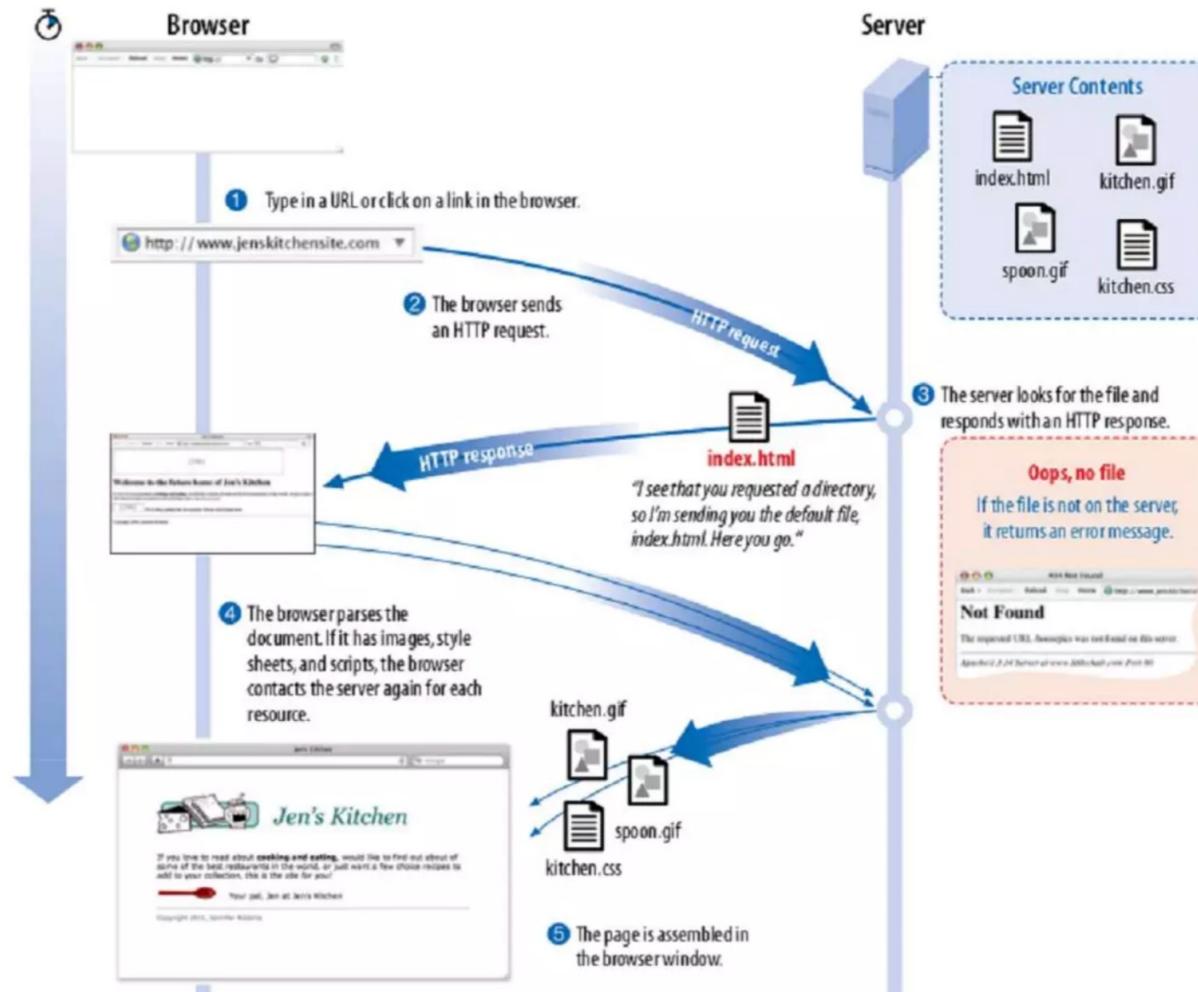


<https://www.slideshare.net/randyconnolly/chapter01-presentation-16514220>

# 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").
- **Dynamic Content Generation:** Modern web servers can also generate dynamic content. They use server-side scripts and databases to create customized responses based on the user's request or session information. This allows for personalized pages that reflect user preferences or real-time data.

# Web Browsers

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

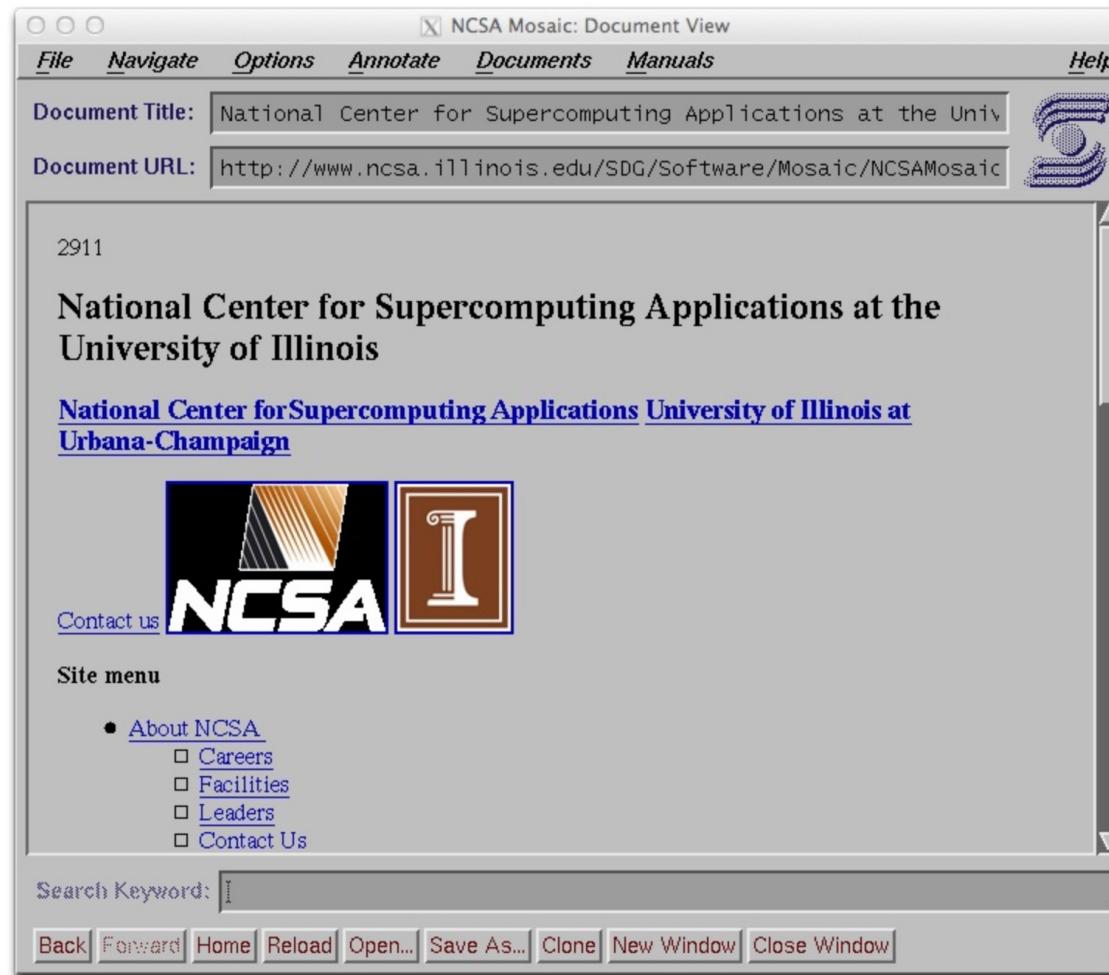
- Chrome, Edge, Safari, Firefox, Chrome, Opera...
- All can handle HTML and HTTP/S
- HTML, CSS, JavaScript

**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)

The image shows a split-screen view of a web browser. On the left, the browser's developer tools or a code editor displays the source code of an HTML file. On the right, the rendered output of the same code is shown in a browser window.

**Source Code (Left):**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Let's Style this Web Page</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            font-size: 16px;
            color: #333;
        }

        p {
            text-align: center;
            margin: 20px 0;
        }

        .content-box {
            border: 2px solid #000;
            padding: 20px;
            background-color: #f0f0f0;
        }

        #main-heading {
            color: #007BFF;
            font-size: 24px;
        }
    </style>
</head>
<body>
<div class="content-box">
    <h1 id="main-heading">Hello, world!</h1>
    <p>Welcome to your first webpage.</p>
</div>
</body>
</html>
```

**Renders (Right):**

The rendered output shows a centered **Hello, world!** heading in a large, blue font. Below it is a paragraph of text: "Welcome to your first webpage."

# 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 (PHP) 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();
?>

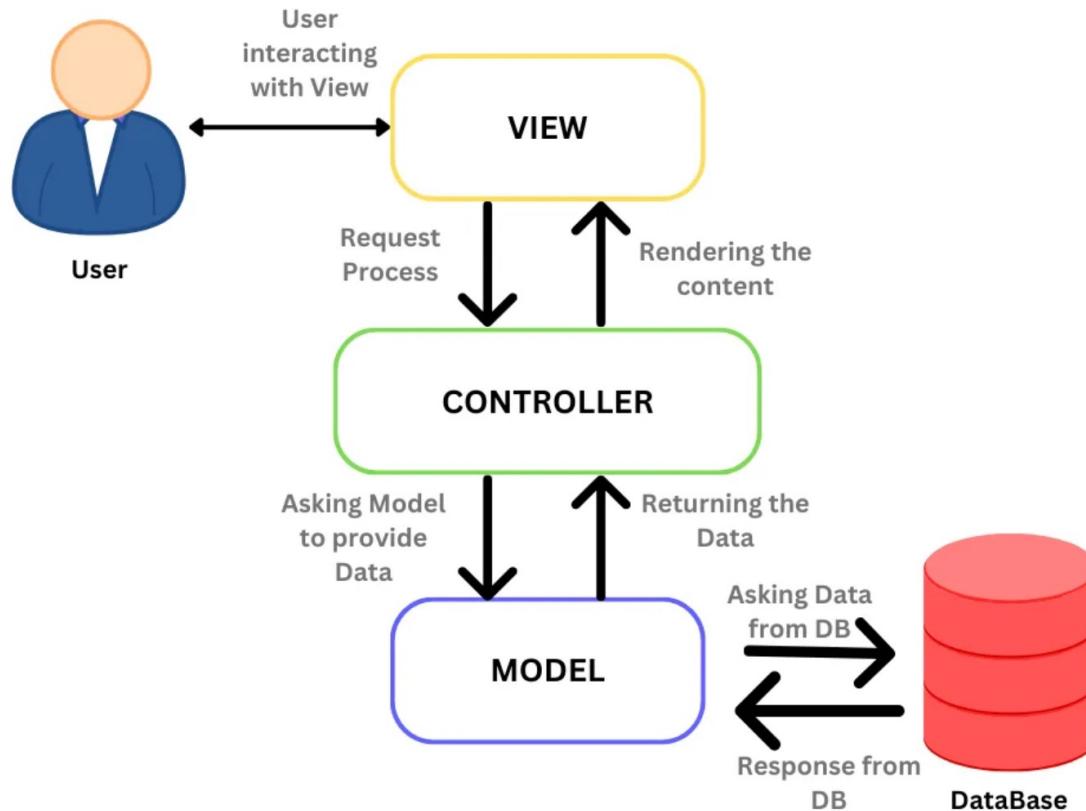
</body>
</html>
```

# 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 CodeIgniter

- **Model (ExampleModel.php):**  
Handles database operations.
- **View (example\_view.php):**  
Contains the HTML and JavaScript that will be sent to the browser.
- **Controller (ExampleController.php):**  
Fetches data from the Model and passes it to the View.

# MVC in CodeIgniter - Model

```
1  <?php
2
3
4  namespace App\Models;
5
6  use CodeIgniter\Model;
7
8  class ExampleModel extends Model
9  {
10     protected $table = 'example_table';
11     protected $primaryKey = 'id';
12     protected $allowedFields = ['name', 'value'];
13
14     public function getAllData()
15     {
16         return $this->findAll();
17     }
18 }
19
```

# MVC in CodeIgniter - Controller

```
2  <?php
3
4  namespace App\Controllers;
5
6  use App\Models\ExampleModel;
7
8  class ExampleController extends BaseController
9  {
10     public function index()
11     {
12         $model = new ExampleModel();
13
14         $data = [
15             'data' => $model->getAllData(),
16         ];
17
18         echo view('example_view', $data);
19     }
20 }
21
```

# MVC in CodeIgniter - Model

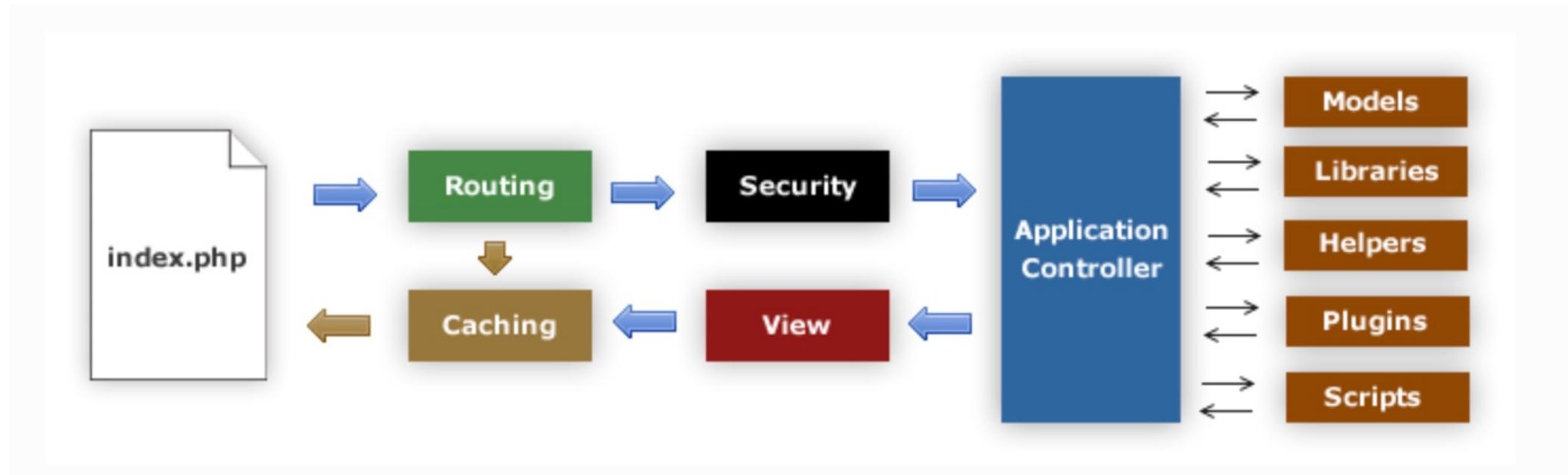
```
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <title>PHP-HTML-JS MVC Example</title>
6      <script type="text/javascript">
7          function toggleTable() {
8              var table = document.getElementById("dataTable");
9              table.style.display = table.style.display === "none" ? "table" : "none";
10         }
11     </script>
12 </head>
13 <body>
14
15     <button onclick="toggleTable()">Show/Hide Table</button>
16
17     <?php if (!empty($data) && is_array($data)): ?>
18         <table id="dataTable" style="display: none;">
19             <tr>
20                 <th>ID</th>
21                 <th>Name</th>
22                 <th>Value</th>
23             </tr>
24             <?php foreach ($data as $row): ?>
25                 <tr>
26                     <td><?= esc($row['id']); ?></td>
27                     <td><?= esc($row['name']); ?></td>
28                     <td><?= esc($row['value']); ?></td>
29                 </tr>
30             <?php endforeach; ?>
31         </table>
32     <?php else: ?>
33         <p>No results found.</p>
34     <?php endif; ?>
35
36     </body>
37 </html>
38
```

# MVC in CodeIgniter - Routes

```
$routes->get('example', 'ExampleController::index');
```

- A Route maps a URL path to the controller.
- Get is just a web page request.
- In the course you will learn about post, which accepts form field values.

# CodeIgniter – PHP MVC Web Framework



<https://codeigniter.com/userguide3/overview/appflow.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.

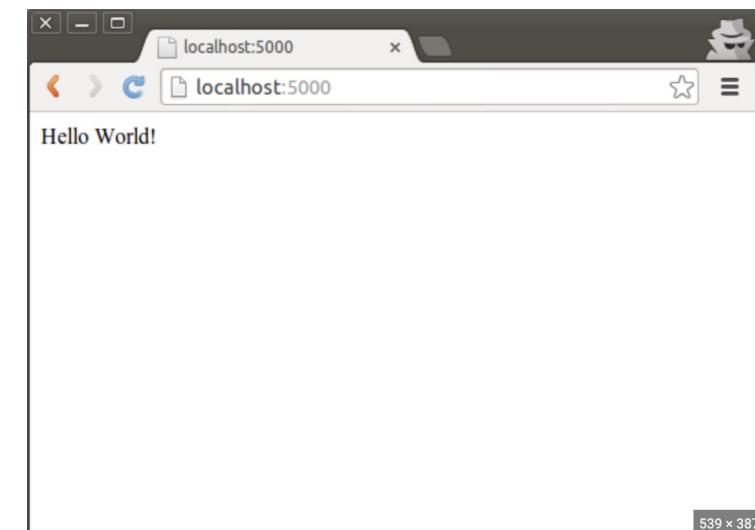
# MVC in Flask

- a popular lightweight web framework for Python that is used to build web applications quickly and easily.
- simplicity and flexibility for both beginners and experienced developers.
- highly extensible, with a wide range of third-party extensions available to add functionality to web applications (e.g. user authentication and security to database integration and content management).
- any size of web application (from small personal projects to large enterprise-level applications)
- thanks to Python and ML strengths, data scientists use it as a web interface.
- Other comparable frameworks: Laravel, Django, Next.js
  - rapid development
  - easy to learn

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello() -> str:
    return "Hello World"

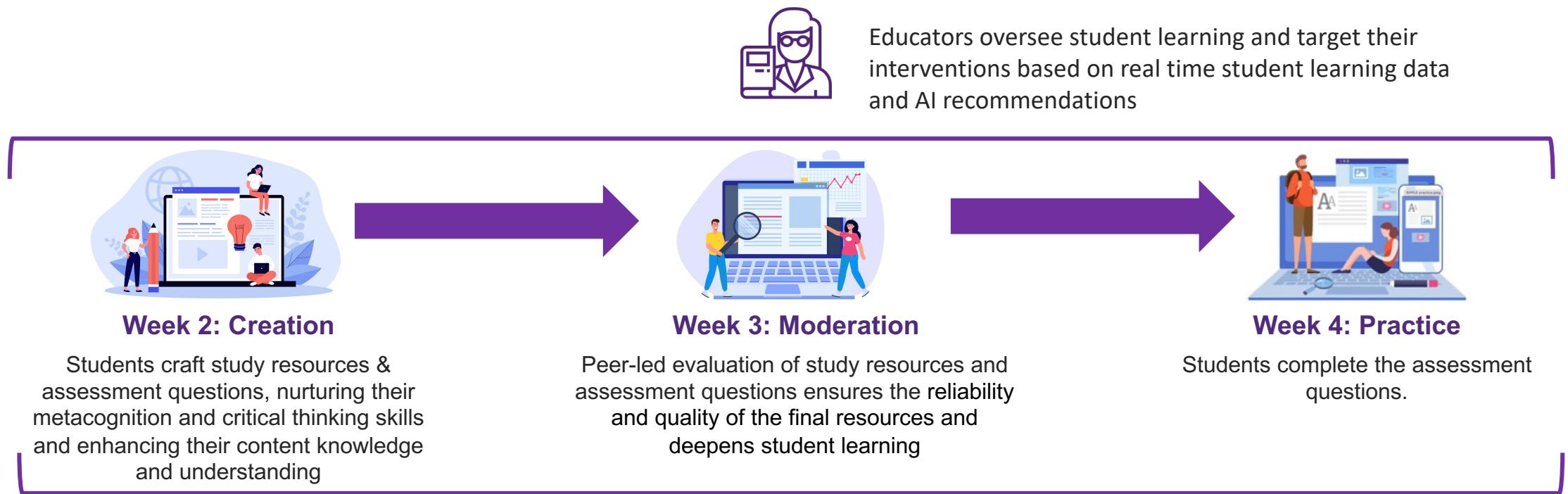
if __name__ == "__main__":
    app.run(debug=False)
```



# Week 1: Todo

- Practicals start in Week 2
- Weekly RiPPL E task is due in Week 2 on Friday at 3pm but the task is open now and you can complete it
- Complete the Self-paced HTML and CSS tutorials if you need a refresh

# 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

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 GenAI Ethically and Creatively
  - The Web is a great way to distribute your application



# Q&A



CREATE CHANGE

# Thank you