

# INTRODUCTION

INTERNET ENGINEERING

Fall 2022

@1995parham

# PARHAM ALVANI

1995parham.github.io



parham

PARHAM [DOT] ALVANI [AT] GMAIL [DOT] COM

NO INSTANT MESSAGING  
**EMAIL** IS AWESOME



# THIS COURSE

- **Web**: the major Internet technology-in-use
  - What are/were the problems?
  - How do/did we solve them?
    - Which protocols and languages are used
- Engineer approach
  - Understand existing concepts & technologies
    - In class
  - Try examples & sample projects
    - By yourself

# THE WORLD-WIDE WEB

- Original idea (Tim Berners-Lee, 1989)
  - Public information sharing on Internet
  - Hypertext
    - Documents are text which can be displayed/converted to desired output
    - Documents can be linked to each others: Web!
- WWW: A system of interlinked hypertext
  - Now, much more complex/interesting applications

# HOW DOES WWW WORK?

- Client-Server mechanism
- Web servers: Process client's requests
  - File (text, image, video, ...) retrieval requests
  - Computation/Processing (DB lookup, transaction, ...) requests
- Web clients: Send the requests
  - Browser: Interacts with client, Requests for server, Processes and displays response (rendering)
  - Other applications
    - Search engines crawlers
    - Use server as a processing element (distributed computing)
    - ...

# WHAT DO WE WANT TO LEARN?

- How does Gmail work?
- Login (keep me signed in)
- Show emails
- Read/Delete emails
- Refresh the list of emails
- Interactive menus
- Per user customizations (themes)
- ...

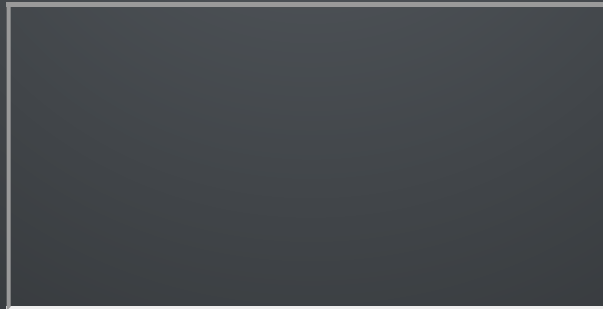


# 10 ENGINEERING QUESTIONS

- Q1) How do web server and client browser talk to each other?
- Q2) How is a web page organized (components)?
- Q3) How is presentation of web page described?
- Q4) How does web page interact with users?
- Q5) How to update a portion of web page?
- Q6) How is transferred data between server & client encoded?
- Q7) How does server process client's requests?
- Q8) How are complex/big web applications developed?
- Q9) How does Gmail offline work?
- Q10) How can other applications use Gmail?

# WWW: FROM OLD TO NOW

- **Static** Web Pages
- Client requests a document from server
  - A communication protocol: **HTTP**
- How to display the document in browser?
  - Document structure definition language: **HTML**
  - Representation of document: **CSS**
  - Later, very later, some advance features: **HTML5**



# WWW: FROM OLD TO NOW

- **Interactive** and **Dynamic** web page
- Needs to interact with user (e.g., event handling in web pages)
  - A programming language in browser: **JavaScript**
- Dynamic data from server (e.g., search result)
  - A programming language in server: **Go**
- Interactive & Dynamic web page
  - A communication mechanism between web page and server: **JavaScript** & **JSON**





# WWW: FROM OLD TO NOW

- Complex processing in server side
  - So many common requirements
    - Threading, Concurrency, Security, etc.
  - Needs an application development framework
    - Web Applications Architectures
- Distributed computing over web
  - Machine-to-Machine communication
    - Function invocation over web
  - Needs a common protocols/API (e.g., Facebook API)
    - Web services

```
{"name": "Luke Skywalker", "height": "172", "mass": "77", "hair_color": "blond", "skin_co
```

# WHAT WE ARE GOING TO LEARN AT CLASSES

# WHAT WE ARE GOING TO LEARN AT CLASSES

- Introduction

# WHAT WE ARE GOING TO LEARN AT CLASSES

- Introduction
- HTTP



# WHAT WE ARE GOING TO LEARN AT CLASSES

- Introduction
- HTTP
- HTML

# WHAT WE ARE GOING TO LEARN AT CLASSES

- Introduction
- HTTP
- HTML
- CSS

# WHAT WE ARE GOING TO LEARN AT CLASSES

- Introduction
- HTTP
- HTML
- CSS
- JavaScript

# WHAT WE ARE GOING TO LEARN AT CLASSES

- Introduction
- HTTP
- HTML
- CSS
- JavaScript
- JSON

# WHAT WE ARE GOING TO LEARN AT CLASSES

- Introduction
- HTTP
- HTML
- CSS
- JavaScript
- JSON
- Golang

# WHAT WE ARE GOING TO LEARN AT CLASSES

- Introduction
- HTTP
- HTML
- CSS
- JavaScript
- JSON
- Golang
- Web Applications

# WHAT WE ARE GOING TO LEARN AT CLASSES

- Introduction
- HTTP
- HTML
- CSS
- JavaScript
- JSON
- Golang
- Web Applications
- Virtualization

# WHAT WE ARE GOING TO LEARN AT TA'S CLASSES



# WHAT WE ARE GOING TO LEARN AT TA'S CLASSES

- Git

# WHAT WE ARE GOING TO LEARN AT TA'S CLASSES

- Git
- HTTP Client

# WHAT WE ARE GOING TO LEARN AT TA'S CLASSES

- Git
- HTTP Client
- HTML/CSS

# WHAT WE ARE GOING TO LEARN AT TA'S CLASSES

- Git
- HTTP Client
- HTML/CSS
- JavaScript/React

# WHAT WE ARE GOING TO LEARN AT TA'S CLASSES

- Git
- HTTP Client
- HTML/CSS
- JavaScript/React
- Backend Development

# WHAT WE ARE GOING TO LEARN AT TA'S CLASSES

- Git
- HTTP Client
- HTML/CSS
- JavaScript/React
- Backend Development
- Docker

# WHAT THIS COURSE IS NOT

- XYZ programming language course for web
  - Many technologies for web development
    - HTML, CSS, XML, etc.
    - Many programming languages: PHP, JS, Golang, etc.
  - You are already programmer
    - You know most programming concepts
    - You need to learn new syntax & new features
- In depth & in detail technology review
  - There are so many technologies

# COURSE ADVANTAGES

- We study and understand technologies that are used in real life every day
  - We don't discuss about pure scientific problems
- An engineering course
  - These technologies are used in industry
  - Better resume: HTML, JavaScript, JSON, CSS, Golang, React, ...
  - More job opportunities (more money 💰💰)
- (Usually) Technologies are easier than sciences
  - No difficult concepts
  - High course grade if you want 😎



# COURSE POSSIBLE DISADVANTAGES

- We study technologies
  - Technologies have limited life time
  - Our knowledge will expire
    - Some programming languages & technologies may not be used 10 years later
  - However, most discussed technologies in this course (hopefully) will have very long life time
- Web development needs many technologies
  - We need to learn many things

# ASSUMPTIONS ON YOUR KNOWLEDGE & SKILLS

- Basic networking concepts
  - Protocol, Port, Header, ...
- Programming
  - Java/C
- Database
  - SQL
- Love to program
  - At least, you don't hate 🙄

# COURSE POLICIES

- There is no grading by TAs (AUT University Policy)
- Course homepage
  - Course slides, Announcements, Grades, etc.
- Grading
  - Midterm + Final + Assignments

# COURSE POLICIES: GRADING 🥲

## Minimum requirement

Midterm	35%	> 40%
Final	35%	> 40%
Homework	30%	- Deadlines wont be extended

# **COURSE POLICIES: GRADING** 😊

# COURSE POLICIES (CONT.)

- If this course is an optional course for you
  - Please don't take it if
    - Your programming skill is poor
    - You assume that it is passed without doing anything
    - You know the answers of almost the questions
      - But you are so busy to do homework
  - You need to allocate enough time for top grades
- If this course is mandatory
  - Note that course topics is a bit wide

# WEB DEVELOPMENT: LECTURES





# WEB DEVELOPMENT: HOMEWORK





# WEB DEVELOPMENT: INDUSTRY



# WEB DEVELOPMENT: GOOGLE, FACEBOOK, ...



# THE MAIN GOAL OF THE COURSE

LEARN AND ENJOY WEB DEVELOPMENT

