



Middlesex
University
London

Computer Science BSc

Dr Luca Piras

Outline

- These slides (and material presented today, e.g. source code) are available at: <https://github.com/PirasLuca/OpenDays-PetStoreApp>

- **Overview of Computer Science BSc**
 - Placement Year (or Studying abroad)
 - Project-Based Approach
 - Overview of Resources and Labs
 - Careers
- **“Web Development” and “Web-Based Mobile App Development”**
 - Example (Demo) from “Web-Based Mobile App Development” 3rd Year Module
 - (Optional) practical experience based on the example
- **Q&A** (at the end but also during the session)

Computer Science BSc: Overview

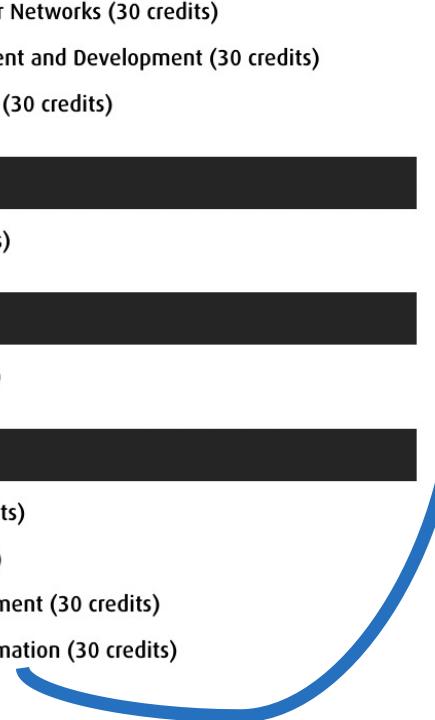
- 3 years
- Full-time
- 4 years with placement
- Many **Project-Based Modules**
- **Wide range of topics** (for instance from software development to systems engineering)
- Right Computer Science **Skills for Industry and Academic Job Opportunities**

Modules
BSc modules
Year 1 - Compulsory
<ul style="list-style-type: none">➢ Programming (30 credits)➢ Systems and Architecture (30 credits)➢ Foundations of Computer Science (30 credits)➢ First Year Project (30 credits)
Year 2 - Compulsory
<ul style="list-style-type: none">➢ Object-Oriented Programming (30 credits)➢ Operating Systems and Computer Networks (30 credits)➢ Software Engineering Management and Development (30 credits)➢ Web Applications and Databases (30 credits)
Placement Year - Optional
<ul style="list-style-type: none">➢ Industrial Placement (120 credits)
Year 3 - Compulsory
<ul style="list-style-type: none">➢ UG Individual Project (30 credits)
Year 3 - Optional
<ul style="list-style-type: none">➢ Testing and Verification (30 credits)➢ Business Intelligence (30 credits)➢ Web-Based Mobile App Development (30 credits)➢ Enterprise Networking and Automation (30 credits)

Did you know

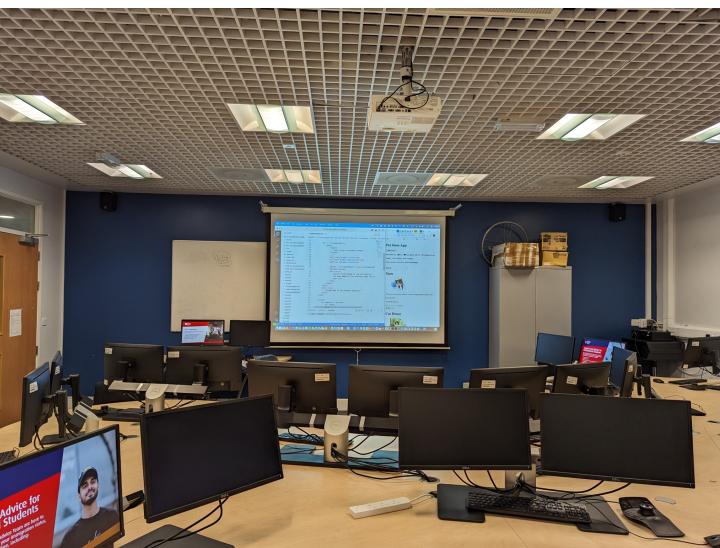
We're ranked 176-200 in the world

for computer science (THE World University Ranking, 2020).

- 
- Digital Media Technology (30 credits)
 - UX Design (30 credits)
 - Artificial Intelligence (30 credits)
 - Designing Secure Systems (30 credits)
 - Computer Graphics (30 credits)
 - Novel Interaction Technologies (30 credits)
 - Advanced Web Development with Big Data (30 credits)
 - Teaching Computing in the Secondary School (30 credits)
 - Systems Engineering for Robotics (30 credits)

Computer Science BSc: Labs and Resources

Studying computing at Middlesex University



Modules

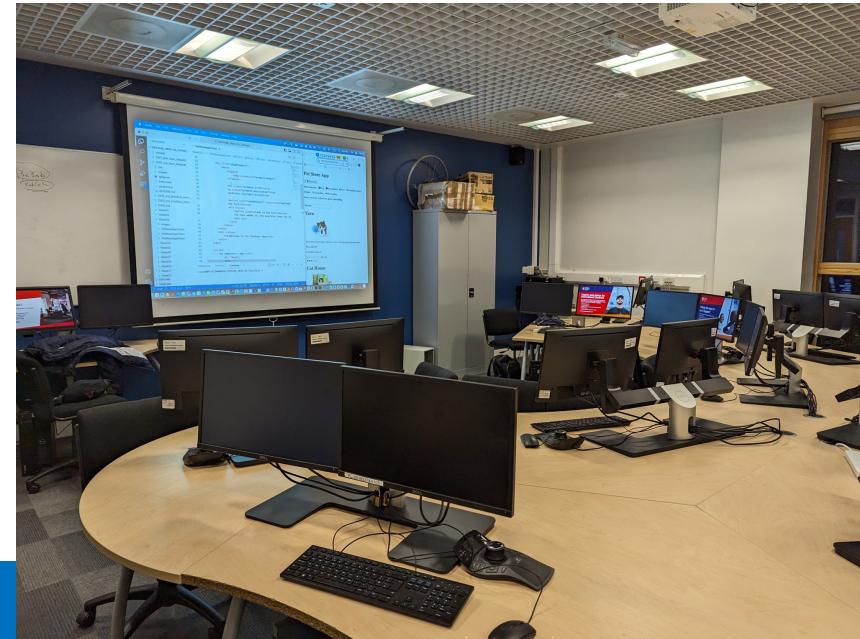
BSc modules

Year 1 - Compulsory

- Programming (30 credits)
- Systems and Architecture (30 credits)
- Foundations of Computer Science (30 credits)
- First Year Project (30 credits)

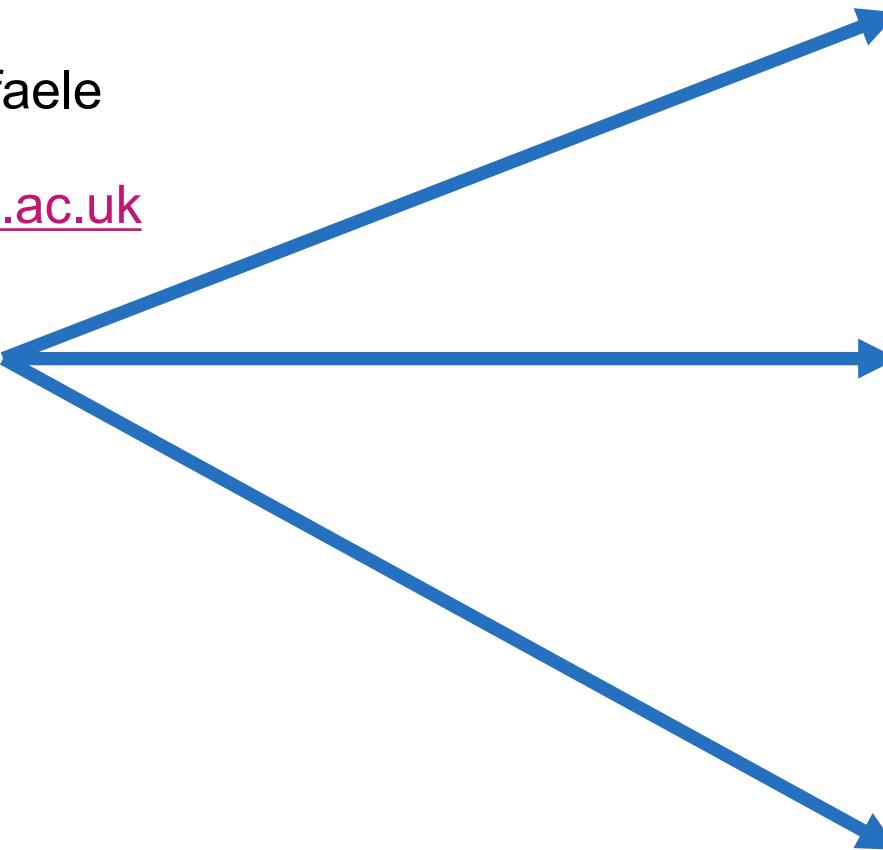
Project-Based approach:

- Foundations of Computer Science
- Systems and Architecture
- Robotics
- ...



- More Detailed Questions

- Full details on Computer Science BSc: [link](#)
- Dr Clifford De Raffaele (Course Leader)
c.deraffaele@mdx.ac.uk
- Admissions: [link](#)



Contact Middlesex University London

Our address

Middlesex University
The Burroughs
Hendon
London NW4 4BT

[Directions to our London campus >](#)

[How to visit our campus >](#)

Prospective students from the UK

Our admissions enquiries team is happy to answer all of your questions about Middlesex and can support you throughout the admissions process. They are available to provide personalised advice, Monday to Friday, 9am to 5pm (except bank holidays, and Easter and Christmas vacations).

[Call +44 \(0\)20 8411 5555 >](#)

[Email us via our contact form >](#)

[Live chat to an adviser >](#)

[Explore your prospectus >](#)

Prospective international students

We have an international network of Regional Offices across the globe dedicated to advising and supporting international students on all aspects of their application to Middlesex.

[Find your nearest regional office >](#)

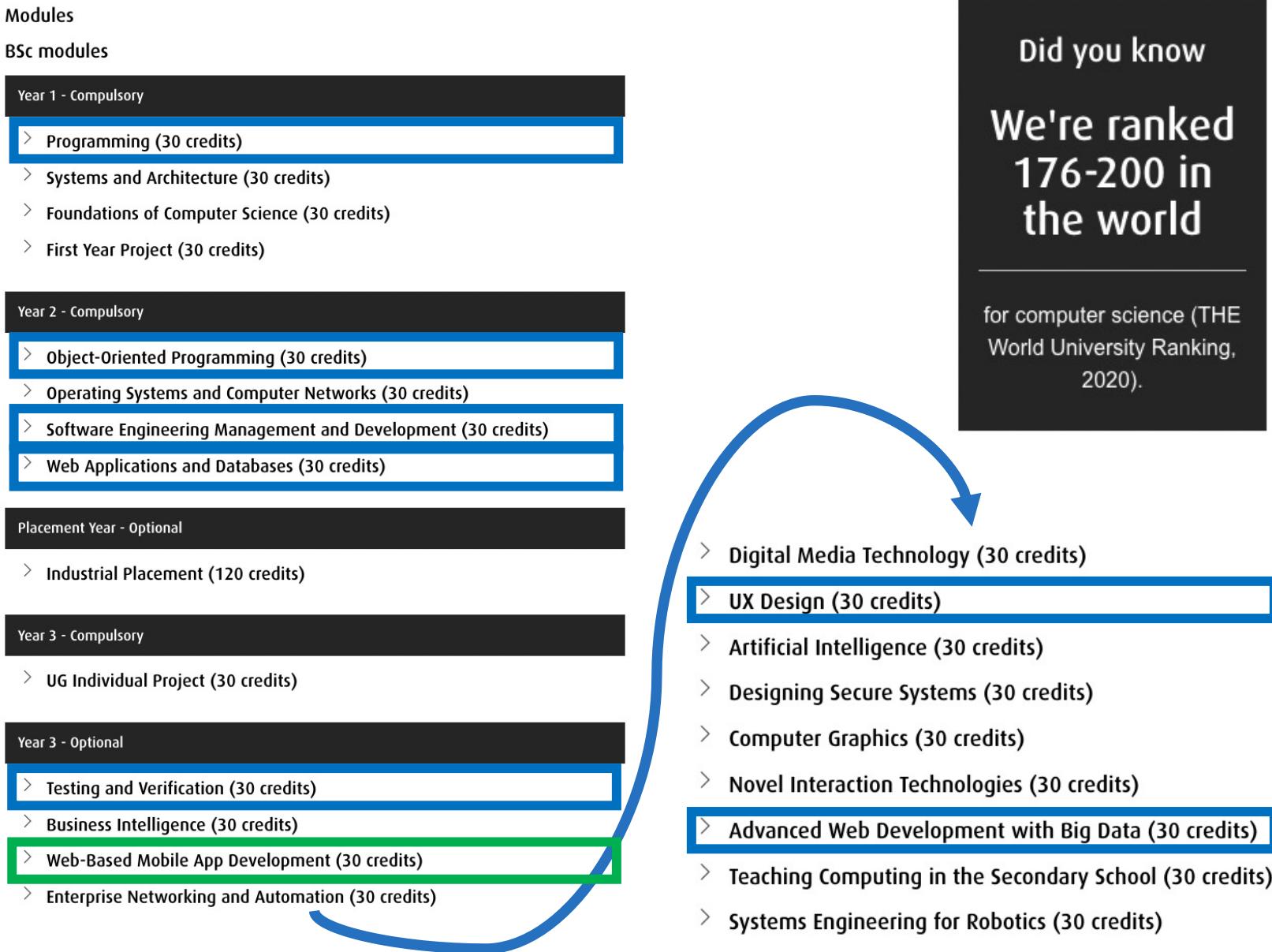
Where to next (after this session)?

Make sure to drop by your **Faculty Lounge** in the **Quad (main building)** or **Grove for Arts and Creative Industries**, after your session. **Meet senior members of the faculty** and speak to some of our **course ambassadors** to **find out more about studying at MDX**. You'll find out more about what it's like to be part of the **MDX community**, get more information about what your learning will look like and it's your chance to **ask any questions you may still have**.

Web Development and Web-Based Mobile Development

Programming Modules

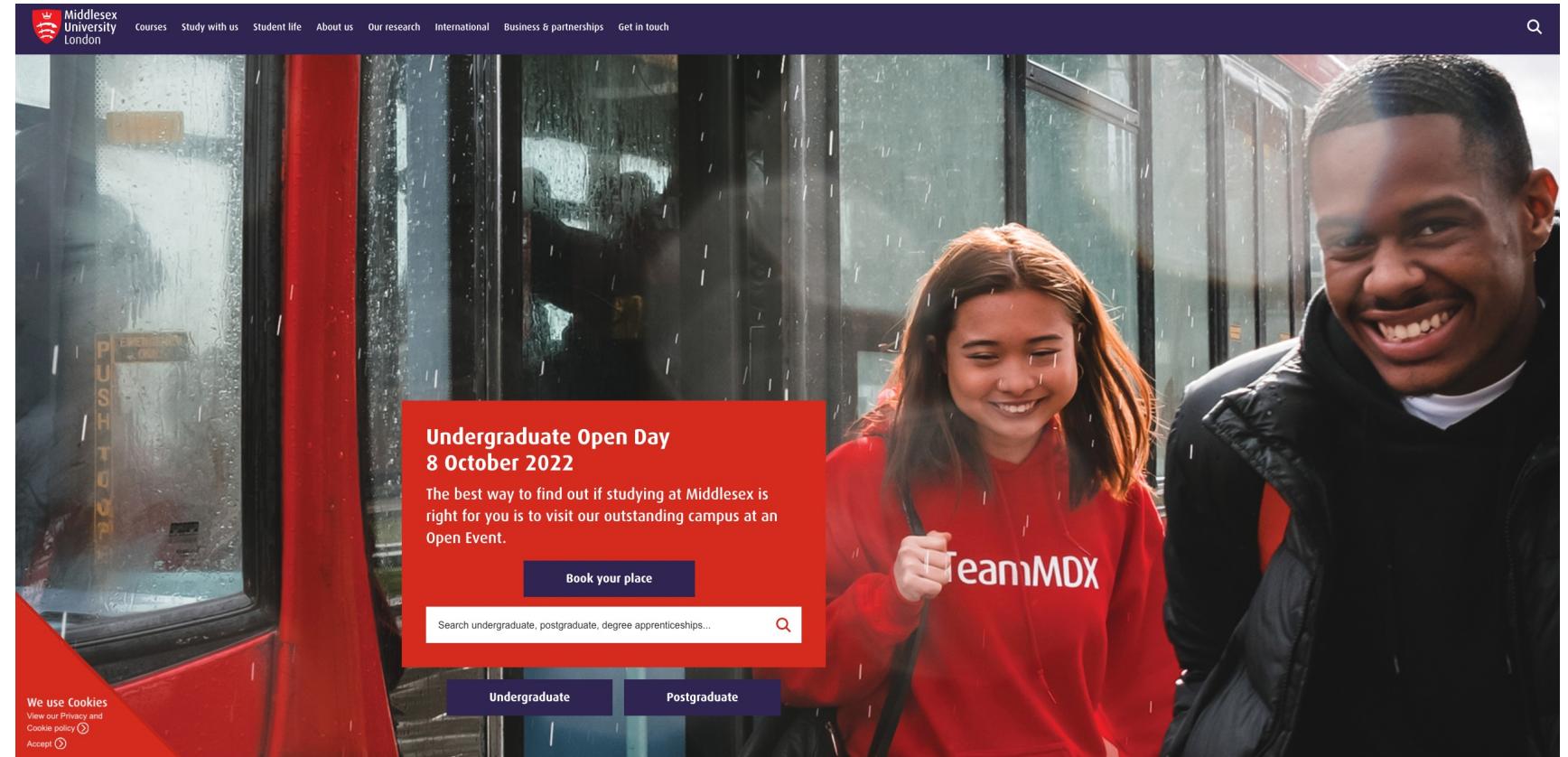
- Software Developer Career
- Software Modules ranging from **basic programming** to **Advanced Web Development with Big Data and Web-Based Mobile App Development**
- Today we will explore:
 - **Web Development**
 - **Web-Based Mobile App Development**
- Today just very basic aspects



Background: Web Development and Web-Based Mobile Development

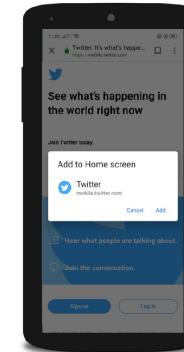
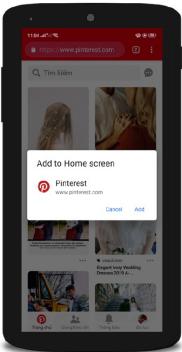
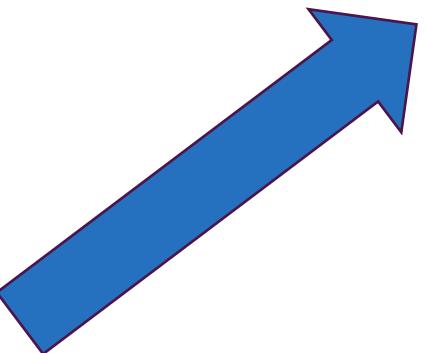
Web Development

- Would you like to explore what there is **behind the creation of a web page?**
- There is a **complex world made of:**
 - Programming languages
 - Tools
 - Programming Environments
 - Heterogeneous Technologies Integrated Together



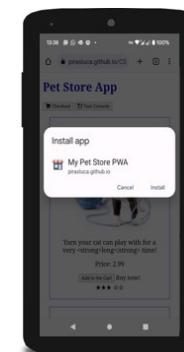
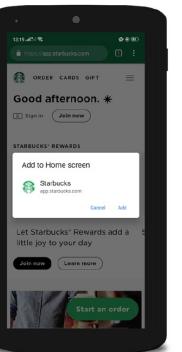
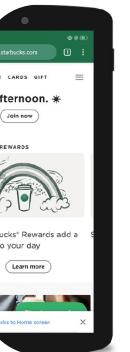
- **Nowadays web-development is becoming even more important**, because your application should run in **multiple platforms** (e.g., web, mobile, smart tv, ...)
- **Good news:** if you develop your app **once**, with the techniques we teach at **MDX**, you will be able to make the app **to run in multiple platforms**

- Hybrid App Development is:
 - **mature and well-established**
 - **widely used**
- Many of the **most used apps are hybrid apps**:
 - Facebook
 - Instagram
 - AirBnB
 - Uber
 - Discord
 - Skype
 - Pinterest
 - Twitter Lite
 - Starbucks
 - ...



Pinterest

Twitter Lite



Starbucks

Our Final App
(Project-Based Approach)

Approach

- **Today** we will be able to see the **very initial basics of Web Development** with an **example**
-> similar also to the **project-based approach** we use at Middlesex University
- From my CST3145 Web-Based Mobile App Development Build an **online pet accessory shop 'Pet Depot'**
- The source code and these slides are available at: <https://github.com/PirasLuca/OpenDays-PetStoreApp>

- Project: E-Commerce App related to selling Pet Items/Food

Vue.js Pet Depot

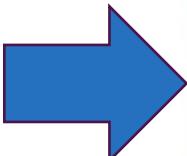


Cat Food, 25lb bag

A 25 pound bag of *irresistible*, organic goodness for your cat.

\$20.00

- This simple extract represents 1 product



Vue.js Pet Depot



Cat Food, 25lb bag

A 25 pound bag of *irresistible*, organic goodness for your cat.

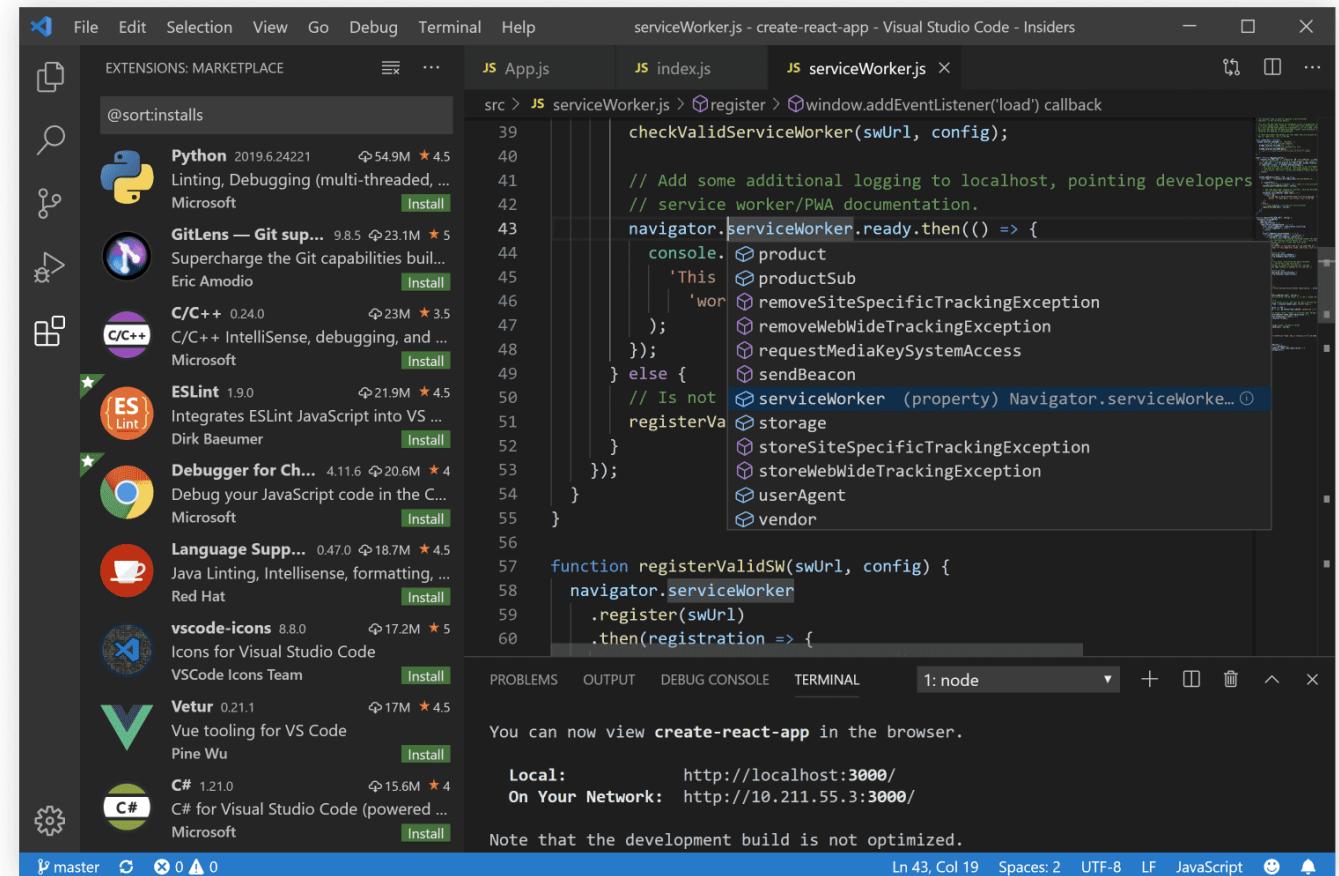
\$20.00

Pet App Project

Let's Start

Integrated Development Environment (IDE): Visual Studio Code

- Used in professional settings by developers
- Ready-to-use functionalities and tools that support the developer in complex projects:
 - To speed up the coding
 - Visual Aids
 - Code Formatting
 - Debugging
 - ...
- Visual Studio Code:
 - Free and Open-Source
 - Made by Microsoft
- First Step:
- Download and Install it from [Official Link](#)



The screenshot shows the Visual Studio Code interface. On the left, the Extensions Marketplace is open, displaying a list of extensions for various programming languages and tools. The extensions listed include Python, GitLens, C/C++, ESLint, Debugger for Chrome, Language Support, vscode-icons, Vetur, and C#. The main code editor on the right shows a file named 'serviceWorker.js' with code related to service workers. The code includes imports for 'register' and 'window.addEventListener', and logic for handling service worker registration and validation. The status bar at the bottom provides information about the file path ('serviceWorker.js - create-react-app - Visual Studio Code - Insiders'), the local host ('http://localhost:3000/'), and the current file statistics ('Ln 43, Col 19').

Let's Start Using Visual Studio Code (VSC) and Some Extensions

- Extensions for Visual Studio Code
 - **Live Server by Ritwick Dey**: (Extension ID: `ritwickdey.LiveServer`) this is very good, mature but not lightweight, so we do not use this, but the next one;
 - **Live Preview by Microsoft**: (Extension ID: `ms-vscode.live-server`) this is less mature, but lightweight, we will use this in some cases
- Emmet (and its docs) is already integrated in VSC; in a html file let's try for instance:
 - type `!` and press enter
 - `input+button+ul>li*3` and press enter; or the following and press enter
 - `div#App>h1{My TodoList}+input#newTask+button#addButton{add}+ul#tasklist>li{Task}*3`
- HTML for our App: `div#App>h1{site name}+main>figure{img}+h1{title}+p{description}+p{Price: }`

Further Guidelines for VSC and Extensions Usage

- **Command Palette**, from the top menu bar "View -> Command Palette" (or "**ctrl shift p**") depending on the system you are using (mac, linux, windows)
- from command palette, if we write **live preview** we can start it
- In the extension settings of "live preview" you can set if to update what you are viewing continuously or only when saving changes

Creating Simple Web Pages with Vue.js

Vue.js Pet Depot



Cat Food, 25lb bag

A 25 pound bag of **irresistible** organic goodness for your cat.

Price: 2000

Adding and Managing Data in the Vue Instance

```
<html>
  <head>
    <title>Vue.js Pet Depot</title>
    <script src="https://unpkg.com/vue@2.7.8/dist/vue.js"></script>
  </head>
  <body>
    <div id="app">
      <header>
        <h1 v-text="sitename"></h1>
      </header>
    </div>
    <script type="text/javascript">
      var webstore = new Vue({
        el: '#app', // <==== Don't forget this comma
        data: { // the 'data' option
          // the key 'sitename' matches the value of 'v-text' earlier
          sitename: 'Vue.js Pet Depot'
        }
      });
    </script>
  </body>
</html>
```

Adding Data to Vue.js for a Product

Add all the information of our product into **data**:

```
data: {  
  sitename: "Vue.js Pet Depot",  
  product: {  
    id: 1001,  
    title: "Cat Food, 25lb bag",  
    description: "A 25 pound bag of <em>irresistible</em>," + "organic goodness for your  
cat.",  
    price: 2000,  
    image: "images/product-fullsize.png"  
  }  
}
```

How to Display the Information in our App

- **Binding the image** with its URL from the data
- **Binding the other information** to be displayed
- Notice the usage of the “Mustache” syntax {{ property-name }}

```
<div id="app">
  <header>
    <h1 v-text="sitename"></h1>
  </header>
  <main>
    <figure>
      <!-- bind the 'src' attribute to the 'product.image' in 'data' -->
      
    </figure>
    <h2 v-text="product.title"></h2>
    <p v-text="product.description"></p>

      <!-- The double curly brackets is the shorter version of 'v-text' -->
      <p>Price: {{product.price}}</p>
    </main>
</div>
```

The Result

We are now displaying the image and all the other information of our product

Vue.js Pet Depot



Cat Food, 25lb bag

A 25 pound bag of **irresistible** organic goodness for your cat.

Price: 2000

Further Activities

- Change the **data** variable and see what happens in the web page
- Inspect **webstore** variable from the console (right click -> “Inspect” -> Console) and change the **data** variable and see what happens in the web page
- Add style to part of the html, for instance:
 - `<p v-html="product.description" style="color:rgb(9, 255, 0);"></p>`
- Play with it as you prefer, ask to the Lecturer and Tutors for support/suggestions

Questions?