**ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ MODULE CSYM019:**

**Internet Programming**  
Programme: MSc Computing (Internet Technology & Security)

**ΔΑΜΙΑΝΑΚΗΣ ΠΟΛΥΒΙΟΣ**

ID Number: 23858016

# **Task 1**

Στην **πρώτη σας εφαρμογή**, θα δημιουργήσετε μια «ζωντανή» λίστα μαθημάτων, όπου θα δημιουργήσετε ένα αρχείο δεδομένων JSON με δυνατότητα αποθήκευσης δεδομένων μαθημάτων. Στη συνέχεια, θα χρησιμοποιήσετε τη Javascript για να εξαγάγετε δεδομένα από το αρχείο δεδομένων JSON και να επεξεργαστείτε και να εμφανίσετε τα δεδομένα σε μορφή HTML.

**VIDEO PRESENTATION:**

<https://youtu.be/ikj6vzi4AmI>

**JSON Schema Validator:**

<https://jsoneditoronline.org/indepth/validate/json-schema-validator/>

**Έλεγχος έκδοσης (GitHub):**

https://github.com/poldam/csym019\_assignment\_2024

**Features Implemented:**

* Αρχείο course.json που περιλαμβάνει όλα τα άλλα σχετικά δεδομένα για την περιγραφή ενός Course του πανεπιστημίου του.
* Αρχείο courseschema.json που περιλαμβάνει το σχήμα του προηγούμενου αρχείου course.json
* Tα τέλη εμφανίζονται σε ένα από τα τρία νομίσματα: Λίρα Ηνωμένου Βασιλείου, Ευρώ ή Δολάριο ΗΠΑ.
* H εφαρμογή χρησιμοποιεί τις τρέχουσες ισοτιμίες μετατροπής από Λίρα σε Ευρώ ή από Λίρα σε Δολάριο για να υπολογίσoουμε τις προμήθειες σε Ευρώ και σε Δολάριο αντίστοιχα (Χρήση του <https://www.exchangerate-api.com/>).
* Τα δεδομένα ενημερώνονται αυτόματα σε ένα ρεαλιστικό προκαθορισμένο διάστημα (var RELOADSECONDS). Αυτό επιτυγχάνεται χρησιμοποιώντας την μέθοδο **setTimeout**.
* Δεν υπάρχει ΚΑΜΙΑ ΚΑΘΥΣΤΕΡΗΣΗ στη φόρτωση των δεδομένων κατά το πρώτο άνοιγμα της σελίδας.
* Το παραγόμενο HTML έχει μορφοποιηθεί, συμπεριλαμβανομένης της διάταξης σελίδας, σε επαγγελματικά πρότυπα χρησιμοποιώντας CSS, με βάση το template του Task 2.
* Συμπεριλαμβάνεται ένα *εικονίδιο μαθήματος (.png)* για κάθε αναφερόμενο course.
* Υλοποίηση εμφωλευμένων accordions (<https://www.w3schools.com/howto/howto_js_accordion.asp>).

# **Course.js**

var LESSONS = []; //Array to hold the courses that come from the json fils

var SELECTEDLESSON = 0; // Currently selected course id

var SELECTEDCURR = 'euro';// Currently selected currency

var USDRATE = 1.27; // Deafult USD rating just in case the rates api call fails

var EURORATE = 1.17; // Deafult EURO rating just in case the rates api call fails

var RELOADSECONDS = 100; // Reload time in seconds used in setTimeout function to reload json file

// Function that loads and handles the json file

// Args:

// jsonFile: path to json file

// callback: name of function that serves as the callback.

function loadAndHandleJSON(jsonFile, callback) {

$.ajax({

url: jsonFile, // get file from url

dataType: 'json', // file type is json

success: function(data) { // on success

callback(null, data); // call callback function with data

},

error: function(jqXHR, textStatus, errorThrown) { // in case of error

callback(errorThrown, null); // call callback funtion with error

}

});

setTimeout(function() { // Periodically reload the json file

loadAndHandleJSON(jsonFile, setUpLessons); //load and handle json file using setUpLessons() as the callback function

}, RELOADSECONDS \* 1000); //Reload every RELOADSECNODS seconds

}

// SECOND VERSION WITH fetch() and no jQuery

// function loadAndHandleJSON(jsonFile, callback) {

// fetch(jsonFile) // get json file

// .then(response => response.json()) // turn it into json

// .then(data => callback(null, data)) // pass the data to the callback function

// .catch(error => callback(error, null)); //in case of an error pass the error to the callback function

// setTimeout(function () { // Periodically reload the json file

// loadAndHandleJSON('./course.json', setUpLessons);

// }, RELOADSECONDS \* 1000);

// }

// Function that sets up the lessosn that we load from the file

// Args:

// error: potential error that is thrown during the loading

// lessond: json data in an array

function setUpLessons(error, lessons) {

if (error) { // if there is an error

console.error('Error loading JSON file:', error);

} else { // if data is loaded properly

console.log('JSON file loaded successfully:', lessons);

// Get the reference to the <ul> element

var lessonsList = document.getElementById("lessonsList"); // placeholder to print that data in our html page

lessonsList.innerHTML = ''; // We empty the placeholder

lessons.forEach(function (lesson) { // for each lesson in the array of lessons

// console.log(lesson.id, lesson.title);

LESSONS[lesson.id] = lesson; // save the lesson in the LESSONS dictionary for future use

var li = document.createElement("li");// Create a new <li> element / menu item

li.id = 'lesson-' + lesson.id; // Set the id attribute of the <li> element

li.className = 'lessonlink'; // Set the class attribute of the <li> element

li.textContent = lesson.title; // Set the text content of the <li> element to the current item's name

li.addEventListener("click", lessonClicked); // Add a listener to the menu item that calls lessonClicked()

lessonsList.appendChild(li); // Append the <li> element to the <ul> element

});

if(SELECTEDLESSON) // If there is a lesson selected then trigger a menu item click to show that lesson

document.getElementById("lesson-" + SELECTEDLESSON).click();

}

}

// Function that sets up an accordion

// Args:

// id: a string to define the accordion class suffix

function setupAccordion(id) {

var acc = document.querySelectorAll(".accordion" + id); // select the accordion class based on the function argument

for (var i = 0; i < acc.length; i++) {

acc[i].addEventListener("click", function () { // add click listener to the button

this.classList.toggle("active"); // on click toggle the panel

var panel = this.nextElementSibling; // get next dom elenet, which is the panel

if (panel.style.maxHeight) { // if the element has a height which means it was open,

panel.style.maxHeight = null; // set the height to null => close it

} else {

panel.style.maxHeight = "10000px"; // set max height to 10000 => open the panel

}

});

}

}

// Function to render a lesson with DOM manipulation

// Args:

// event: var that holds all the info of the event that triggered the function

function lessonClicked(event) {

var lessonLinks = document.querySelectorAll(".lessonlink"); // select all elements that have class .lessonlink

lessonLinks.forEach(function (link) { // go through the menu items and remove the lessonlinkActive class that makes the highlighted

link.classList.remove("lessonlinkActive");

});

event.target.classList.add("lessonlinkActive"); // highlight the clicked menu item

var id = event.target.id.replace("lesson-", ""); // Extract the ID of the clicked <li> element

SELECTEDLESSON = id; // set the global id that holds the lesson

var lesson = LESSONS[id]; // select the lesson from the global LESSONS array

var currency = "EUR"; //default currency

var rate = EURORATE; // default rate

if (SELECTEDCURR == "pound") { // if selected currency is pounds

currency = "GBP"; //setup currency and rate

rate = 1;

}

else if (SELECTEDCURR == "dollar") { // if selected currency is dollar

currency = "USD"; //setup currency and rate

rate = USDRATE;

}

var formatter = new Intl.NumberFormat('en-US', { // formatter for the currency

style: 'currency',

currency: currency, // Change to the appropriate currency code

minimumFractionDigits: 0,

maximumFractionDigits: 0

});

// helper vars for holding the html before passing it to a DOM element for rendering

var html = "";

var extras = "";

var header = "";

var credits = "";

// Display the lesson details

var lessonDetails = document.getElementById("lessonDetails"); //Get lesson details placeholder

var lessonKeys = document.getElementById("lessonKeys"); // Get lesson keys placeholder

// Clear the html of those 2 elements

lessonDetails.innerHTML = '';

lessonKeys.innerHTML = '';

if (lesson) { // if there is a lesson

if (lesson.title) // if there is a lesson title set it up

header += "<h2>" + lesson.title + "</h2>";

if(lesson.image) { // if there is a lesosn image set that up

header += "<div class='leftHeader'><div class='square-container'> <img src='./images/" + lesson.id + ".png' alt='Your Image'> </div>";

if (lesson.link) // if there is a lesson link set it up

header += "<p><a class='button roboto-bold' target='\_blank' href='" + lesson.link + "'> Read more </a></p>";

header += "</div>";

header += "<div class='rightHeader'>";

// if there is an overview section set it up

if (lesson.overview)

header += "<h4>OVERVIEW</h4>";

header += "<p>" + lesson.overview + "</p>";

// if there are highlights set them up as a list

if (lesson.highlights) {

header += "<h4>HIGHLIGHTS</h4>";

header += "<ul class='oneempaddingleft'>";

lesson.highlights.forEach(function (item) {

header += "<li>" + item + "</li>";

});

header += "</ul>";

}

header += "</div>";

} else {

// if there is no image setup overview and link in a simple way

if (lesson.overview)

header += "<p>" + lesson.overview + "</p>";

if (lesson.link)

header += "<p><a class='button roboto-bold' target='\_blank' href='" + lesson.link + "'> Read more </a></p>";

}

lessonDetails.innerHTML += header; // add header to the lessons details html

if (lesson.COURSECONTENT) { // if there is a COURSE CONTENT section set it up

lessonDetails.innerHTML += "<hr>";

lessonDetails.innerHTML += "<h3>COURSE CONTENT</h3>";

if (lesson.COURSECONTENT.CourseDetails) { // If there are course details

html = "";

if (lesson.COURSECONTENT.CourseDetails.html)

html = lesson.COURSECONTENT.CourseDetails.html

if(lesson.COURSECONTENT.CourseDetails.stages) { //if there are course stages

lesson.COURSECONTENT.CourseDetails.stages.forEach(function (item) { // for each stage create a new accordiion that will hold the modules

html += "<button class='accordion2'>" + item.name + "</button>"; // click on the button to expand the stage panel

if(item.modules) { // if there are modules inside the stage

item.html = "";

item.modules.forEach(function (item2) { // for each module create a new accordion and manipulate its dom to present the module

credits = ""

if(item2.credits) {

credits = " (" + item2.credits + " Credits)";

}

item.html += "<button class='accordion3'>" + item2.name + credits + " </button>"; //click on the button to show the panel

// Module information

item.html += "<div class='panel3'><br><div class='gray'><small><span class='roboto-bold'>Module code:</span> " + item2.code + " <span class='roboto-bold'>Status:</span> " + item2.status + " </small></div> <br>" + item2.description + "<br><br></div>";

});

}

// stage html

html += "<div class='panel2'><br>" + item.html + "<br></div>";

});

}

// Create an accordion for the module inside the stage accordion

lessonDetails.innerHTML += "<button class='accordion'>Course Details</button>";

lessonDetails.innerHTML += "<div class='panel'><br>" + html + "<br></div>";

}

if (lesson.COURSECONTENT.EntryRequirements) { // if there are EntryRequirements set them up

html = "";

if (lesson.COURSECONTENT.EntryRequirements.html)

html = lesson.COURSECONTENT.EntryRequirements.html

// create an accordion

lessonDetails.innerHTML += "<button class='accordion'>Entry Requirements</button>";

lessonDetails.innerHTML += "<div class='panel'><br>" + html + "<br></div>";

}

if (lesson.COURSECONTENT.FeesandFunding) { // if there are FeesandFunding set them up

html = "";

if(lesson.COURSECONTENT.FeesandFunding.title) // setup title

html +="<h3>" + lesson.COURSECONTENT.FeesandFunding.title + "</h3>";

if(lesson.COURSECONTENT.FeesandFunding.text) // setup text

html +="<p>" + lesson.COURSECONTENT.FeesandFunding.text + "</p>";

lesson.COURSECONTENT.FeesandFunding.Fees.forEach(function (item) { // go through fees array and setup fees

extras = "";

if(item.extra)

extras = item.extra;

html += "<div><span class='roboto-medium'>" + item.type + "</span>: " + formatter.format(Math.round(rate\*item.value)) + " " + extras + "</div>"; //format the fee and readjust its value based on the currency selected. Use the rate value. Round it up

});

if(lesson.COURSECONTENT.FeesandFunding.AdditionalCosts) { // If there are additional costs set it up

html +="<h3>Additional Costs</h3>";

html +="<p>" + lesson.COURSECONTENT.FeesandFunding.AdditionalCosts + "</p>";

}

lessonDetails.innerHTML += "<button class='accordion'>Fees and Funding</button>"; // Create an accordion

lessonDetails.innerHTML += "<div class='panel'><br>" + html + "<br></div>";

}

if (lesson.COURSECONTENT.StudentPerks) { // If there are student pers set them up

html = "";

if (lesson.COURSECONTENT.StudentPerks.html)

html = lesson.COURSECONTENT.StudentPerks.html

lessonDetails.innerHTML += "<button class='accordion'>Student Perks</button>"; // create an accordion

lessonDetails.innerHTML += "<div class='panel'><br>" + html + "<br></div>";

}

if (lesson.COURSECONTENT.IntegratedFoundationYear) { //if there is IFY set it up

html = "";

if (lesson.COURSECONTENT.IntegratedFoundationYear.html)

html = lesson.COURSECONTENT.IntegratedFoundationYear.html

lessonDetails.innerHTML += "<button class='accordion'>Integrated Foundation Year (IFY)</button>"; // create accordion

lessonDetails.innerHTML += "<div class='panel'><br>" + html + "<br></div>";

}

if (lesson.COURSECONTENT.FAQs) { // if there is a FAQs section in json set that up

html = "";

if (lesson.COURSECONTENT.FAQs.html)

html = lesson.COURSECONTENT.FAQs.html; //insert faq html

if(lesson.COURSECONTENT.FAQs.questions) {

lesson.COURSECONTENT.FAQs.questions.forEach(function (item) { // go through array with questions/answers

html += "<button class='accordion2'>" + item.q + "</button>"; //set up accordion for each pair

html += "<div class='panel2'><br>" + item.a + "<br><br></div>";

});

}

lessonDetails.innerHTML += "<button class='accordion'>FAQs</button>"; // setup accordion for FAQs section

lessonDetails.innerHTML += "<div class='panel'><br>" + html + "<br></div>";

}

}

if (lesson.KEYFACTS) { // if there is a KEY FACTS section set it up

lessonKeys.innerHTML += "<h2>KEY FACTS</h2>";

if (lesson.KEYFACTS.UCASCode) { // set up the codes

lessonKeys.innerHTML += "<div class='roboto-bold'>UCAS Code</div>";

lesson.KEYFACTS.UCASCode.forEach(function (item) { // go through the codes array

lessonKeys.innerHTML += "<div><span class='roboto-medium'>" + item.type + "</span>: " + item.value + "</div>";

});

lessonKeys.innerHTML += "<hr>"; // use a separator

}

if (lesson.KEYFACTS.Level) { // if there is a level field set it up

lessonKeys.innerHTML += "<div><span class='roboto-bold'>Level</span>: " + lesson.KEYFACTS.Level + "</div>";

lessonKeys.innerHTML += "<hr>";// use a separator

}

if (lesson.KEYFACTS.Duration) { // if there is a duration field set it up

lessonKeys.innerHTML += "<div class='roboto-bold'>Duration</div>";

lesson.KEYFACTS.Duration.forEach(function (item) {

lessonKeys.innerHTML += "<div><span class='roboto-medium'>" + item.type + "</span>: " + item.value + "</div>";

});

lessonKeys.innerHTML += "<hr>";// use a separator

}

if (lesson.KEYFACTS.Starting) { // if there is a starting field set it up

lessonKeys.innerHTML += "<div><span class='roboto-bold'>Starting</span>: " + lesson.KEYFACTS.Starting + "</div>";

lessonKeys.innerHTML += "<hr>";// use a separator

}

if (lesson.KEYFACTS.EntryRequirements) { // if there is an entry requirements field set it up

lessonKeys.innerHTML += "<div class='roboto-bold'>Entry Requirements:</div><div>" + lesson.KEYFACTS.EntryRequirements + "</div>";

lessonKeys.innerHTML += "<hr>";// use a separator

}

if (lesson.KEYFACTS.Fees) { // if there is a fees section

if (lesson.KEYFACTS.Fees.UK) { // set up fees for UK

lessonKeys.innerHTML += "<div class='roboto-bold'>Fees UK</div>";

lesson.KEYFACTS.Fees.UK.forEach(function (item) { // go through the fees array and set it up

extras = "";

if(item.extra)

extras = item.extra;

lessonKeys.innerHTML += "<div><span class='roboto-medium'>" + item.type + "</span>: " + formatter.format(Math.round(rate\*item.value)) + " " + extras + "</div>"; // use currently selected currency and its rate. Format using the formatter

});

}

lessonKeys.innerHTML += "<br>";

if (lesson.KEYFACTS.Fees.International) { // set up international fees

lessonKeys.innerHTML += "<div class='roboto-bold'>Fees International</div>";

lesson.KEYFACTS.Fees.International.forEach(function (item) { // go through the fees array and set it up

extras = "";

if(item.extra)

extras = item.extra;

lessonKeys.innerHTML += "<div><span class='roboto-medium'>" + item.type + "</span>: " + formatter.format(Math.round(rate\*item.value)) + " " + extras + "</div>"; // use currently selected currency and its rate. Format using the formatter

});

}

lessonKeys.innerHTML += "<hr>"; // use a separator

}

if (lesson.KEYFACTS.Location) { // if there is a location field add that too

lessonKeys.innerHTML += "<div><span class='roboto-bold'>Location</span>: " + lesson.KEYFACTS.Location + "</div>";

}

}

} else {

lessonDetails.innerHTML = "<p>Το μάθημα δεν βρέθηκε.</p>"; // no lesson was found so we print an error message

}

// Set up the three levels accordions

setupAccordion(""); // Course Accordion

setupAccordion("2"); // Stages Accordion

setupAccordion("3"); // Modules Accordion

}

// Function to setup the currently used currencu from the currency dropdown

// Args:

// event: information about the event that triggered the function - currency dropdown

function currencyChanged(event) {

SELECTEDCURR = document.getElementById("currencySelector").value; // get the value of the dropdown and set it as the current currency

if(SELECTEDLESSON) // if there is a lesson already selected reload it to get the new fee values based on the currently selected rate

document.getElementById("lesson-" + SELECTEDLESSON).click(); // trigger a menu it click to load again the currently selected lesson

}

// Function to get USD,GBP,EUR rates from a live API

function getRates() {

//https://openexchangerates.org/account/usage

const apiKey = '11fdd6d5f1274db596c9fb847f5497c2'; // Set up API key

const apiUrl = `https://open.er-api.com/v6/latest?base=GBP&symbols=USD,EUR&apikey=${apiKey}`; // Set up API URL and define base, and which rates we want

fetch(apiUrl) // Fetch exchange rates from the API

.then(response => response.json()) // get response in json

.then(data => { // if data is fetched

if (data.result == 'success') { // if the call was successful

USDRATE = data.rates.USD; //setup live USD rate

EURORATE = data.rates.EUR; //setup live EURO rate

} else {

// If call fails then setup default rates

console.log("Failed to retrieve exchange rates.", data);

USDRATE = 1.27;

EURORATE = 1.17;

}

})

.catch(error => { // If there is an API error set up default rates for USD, EUR

console.error('Error fetching exchange rates:', error);

USDRATE = 1.27;

EURORATE = 1.17;

});

}

var currencySelector = document.getElementById("currencySelector"); // get the dom element that serves as the currency selector

currencySelector.addEventListener("change", currencyChanged); // add an event listener to that element so everyday we select a different value and we click it will call the currecyChanged() function

getRates(); // get Current rates for USD,EUR

loadAndHandleJSON('./course.json', setUpLessons); // load and handle json for file './course.json' and use setUpLessons() as a callback function