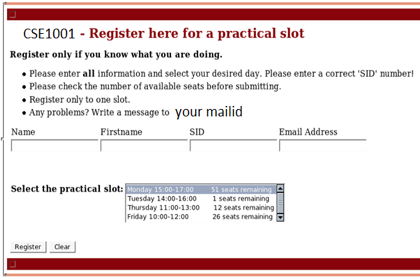
1. There are 32 students assigned to a module. In order to organize module demonstrations, the class should be split into four groups with different time slots. Each group should consist of 8 students. The module lecturer decided to have a registration webpage to allow students to sign up for one of the demonstrations. A student visiting the page should be able to submit her name, student ID, email address, and book a place in one of the time slots. For example, the registration Web page of CSE1001 looks like this:



The submitted data should be stored in a database which is maintained on a server. The webpage and the server should interact with each other at every step of the registration process. The page should show how many free places are available in each time slot, announcing and blocking all fully booked time slots. After a student makes a data submission, it should check whether the student has been already registered. If not, the data is stored on the server and the student is notified about her registration. Otherwise (if already registered), the student should be prompted to ensure that she wants to change her registration to the new section (and removed from the current one she is registered for).

As a means of checking which students are registered in each section, you should also write a **separate** webpage that will allow the module lecturer to choose a section and, after querying the database, will display the list of students who are registered in that section to a webpage.

Write a registration webpage using HTML, CSS, Nodejs/Angularjs, PHP and SQL/NoSQL, as outlined above, in order to allow students to register to demonstration time slots.

1. We want you to write clean and standards compliant PHP to save user information captured from a form into a database, then retrieve and show all records in a list. You may use any framework of your choice or write the assignment in core PHP.

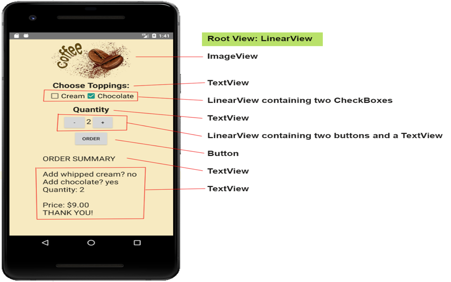
The following fields are needed - Name, Country, Email, Mobile Number, About

You, Birthday

There should be a form page to add data, and a paginated list page to view the data. It should also be possible to edit the data for an existing user.

You may use any CSS or nodejs/angularjs frameworks like bootstrap or jQuery to make the pages look nice and easy to use. The PHP, CSS & Script should follow all best coding practices. We recommend that you make maximum use of HTML & CSS techniques for UI.

1. Build the Coffee Ordering app shown below.   
   Assume a single coffee costs Rs.10.00. Charge an additional Rs. 10.00 for chocolate and Rs. 15 for whipped cream, per cup.  
   Use structure shown in the figure to guide you. Note: the action Background is removed from the app.

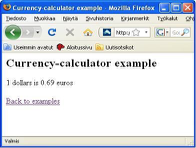
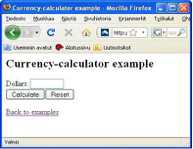


1. Write a PHP program where you take any positive integer n, if n is even, divide it by 2 to get n / 2. If n is odd, multiply it by 3 and add 1 to obtain 3n + 1. Repeat the process until you reach 1.  
   Input : 12  
   Output : Array   
   (   
   [0] => 12  
   [1] => 6   
   [2] => 3   
   [3] => 10  
   [4] => 5   
   [5] => 16  
   [6] => 8   
   [7] => 4   
   [8] => 2   
   [9] => 1   
   )  
   According to Wikipedia the Collatz conjecture is a conjecture in mathematics named after Lothar Collatz, who first proposed it in 1937. The conjecture is also known as the 3n + 1 conjecture.

The conjecture can be summarized as follows. Take any positive integer n. If n is even, divide it by 2 to get n / 2. If n is odd, multiply it by 3 and add 1 to obtain 3n + 1. Repeat the process (which has been called "Half or Triple plus One") indefinitely. The conjecture is that no matter what number you start with, you will always eventually reach 1.

Example :  
For instance, starting with n = 12, one gets the sequence 12, 6, 3, 10, 5, 16, 8, 4, 2, 1.  
n = 19, for example, takes longer to reach 1: 19, 58, 29, 88, 44, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.

1. Implement simple currency calculator utility-program, which can be used to calculate currency conversions from dollars to euros. Exchange rates can be found for example on <http://www.x-rates.com/d/EUR/table.html>. Implement simple HTML-form where user user can input amount of dollars (currency.html) and PHP-script (calculate.php) to calculate currency conversion. PHP-script will read the input values from HTML-form and perform conversion by multiplying dollars with current exchange rate.



1. Design and Develop an Online Resume Maker portal with the following requirements

* Take the input from user in a form of all the fields for a resume
* Clicking on the submit button should verify(validate) data entered in the fields
* Display the resume on the output page with a text field and a send button
* Clicking on send button should write the resume to a text file and email it to the specified email address in the text field

Note: Use PHP, HTML, Nodejs/Angularjs, SQL/NoSQL for the application development.

7. Application using Nodejs

i. Joseph came across the next question, which asked him to check if the word spelt the same backwards .A simple enough question, he started to solve the question only to be constrained by language allowed. (Input a string (of character <80) from the user and print whether the string is a palindrome or not.

Assume no spaces between characters.)

Sample Input: eergreejj

Sample Output: Not At All

(If it is a palindrome - print "Indeed")

SAMPLE INPUT: eergreejj

SAMPLE OUTPUT: Not At All

ii. Build a Node.js server for the following task,

You are given an integer N. You need to print the series of all prime numbers till N.

Input Format: The first and only line of the input contains a single integer N denoting the number till where you need to find the series of prime number.

Output Format: Print the desired output in single line separated by spaces.

Constraints: 1<=N<=1000

SAMPLE INPUT: 9

SAMPLE OUTPUT: 2 3 5 7

8. Add a random "fortune generator" to your home page. That is, your page should contain a list of fortunes (stored as an array of strings), and should randomly select one of those fortunes to display each time the page is loaded. The fortune should be displayed just above the page footer, cantered and enclosed in a box. Here's an example given below.

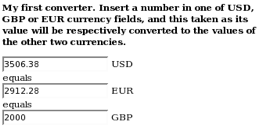
|  |
| --- |
| True wisdom comes not from knowledge, but from understanding. |

Please put your Nodejs Script in an external file, link to it in the <head> element of your page, and just put an appropriate function call on your homepage to display the fortune.

9. You can find the (not-so) current exchange rates amongst several currencies in the table given below. Note that conversions are given by reading down the table. For example, 1 USD = 0.49246 GBP, and 1 CAD = 1.01941 USD.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | USD | GBP | CAD | EUR | AUD |
|  | 1 | 2.03032 | 1.01941 | 1.41544 | 0.88297 |
|  | 0.49246 | 1 | 0.50221 | 0.69714 | 0.43497 |
|  | 0.98054 | 1.99169 | 1 | 1.38814 | 0.86613 |
|  | 0.70641 | 1.43448 | 0.72037 | 1 | 0.62382 |
|  | 1.13262 | 2.29964 | 1.15498 | 1.60329 | 1 |

Create a Web page named convert.php that can be used to convert some value in one currency to the respective values of the other currencies. That is, your page should include a form which consists of five currency fields, and on inserting a number in one of the fields, this number, taken as the respective value of the currency, be converted to the values of the other currencies. For example, a page of USD, EUR and GBP converter might look like this (the converted values don't match the values in the table above):



10. Create a Web page named quiz.php that can be used to conduct multiple choice quizzes over the Web. The page should contain at least 10 potential quiz questions, each with three possible answers (A, B, and C). When loaded, the page should first prompt the person for the number of desired questions in the quiz, with a default of 5 questions. The page should then randomly select questions and prompt the user with each question and possible answers. Each answer entered by the user should be compared with the correct answer, and the result displayed within the page (either CORRECT or INCORRECT). At the end, the number and percentage of correct guesses should be displayed in the page.

Your page must be clear and understandable to the user, and support the following:

* It must be straightforward to add or remove potential questions, with a minimal amount of editing.
* It must display each question and all three potential answers as part of the prompt, and make it clear to the user how their answer should be formatted.
* The question, user's answer, and correctness of that answer must be displayed in the page. In the case of an incorrect answer, the correct answer must be identified.
* The number of correct answers, total number of questions, and correctness percentage must be displayed in a readable format.

For example, the page might contain the following as a result of a 5-question quiz:

What is the capital of Missouri?

    You guessed B) Jefferson City

    CORRECT

    How many ounces in a pound?

    You guessed A) 10

    INCORRECT: the correct answer is C) 16

    Who was the first person to set foot on the moon?

    You guessed C) Neil Armstrong

    CORRECT

    Who holds the Major League Baseball record for most home runs in a season?

    You guessed A) Barry Bonds

    CORRECT

    In what year was University of Liverpool founded?

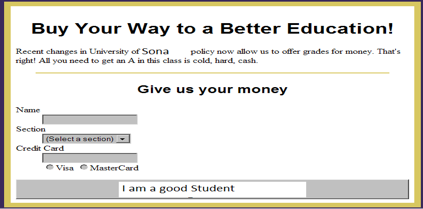
    You guessed A) 1250

    INCORRECT: the correct answer is B) 1881

    You answered 3 out of 5 questions correctly (60%).

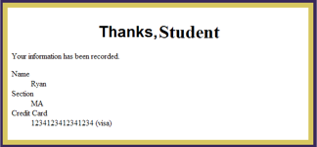
11. Create an HTML form that POSTs its submitted data to a PHP program on a server. Download [buyagrade.html](https://courses.cs.washington.edu/courses/cse190m/09sp/labs/4-buyagrade/buyagrade.html) to your disk (right-click the link and choose Save Link As...). You need to modify this HTML file by turning it into an HTML form. You will need to give name attributes to the form controls so they will be sent to the server as query parameters; the names are up to you. Also, some form controls (such as radio buttons) need value attributes. Fill the Section drop-down list with choices MA through MH.

When you're done with this exercise, your form should look roughly like the screenshot below.

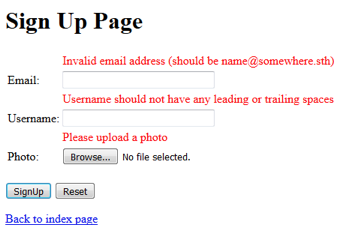


Next, write the PHP page that will handle the submitted form data. Tell you buyagrade.html to POST to student.php. The student.php page will receive the parameters from buyagrade.html and will output an HTML confirmation page. Here is a skeleton version of this page that does not actually display the data submitted by the user. Modify it to display the submitted data.

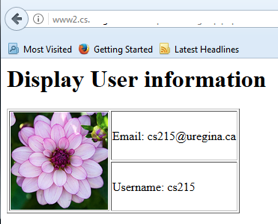
Your page should at least display the submitter's name, credit card number, and credit card type (Visa or MasterCard) in the confirmation page. For now, your page doesn't actually need to save this information in any way on the server.



12. Create an html form [Signup.html](http://www2.cs.uregina.ca/~temp8/lab9/Signup.html) to display the output as shown below



Once the user input the correct information, Create a seperate php file(Signup.php) to upload the image file and display the user information in a table format. Your output should look similar to the following image.



13. i. Write a PHP script to get time difference in days and years, months, days, hours, minutes, seconds between two dates.

Note : Use DateTime class.

ii. Build a Node.js server to say a given string is palindrome or not

input: localhost:8080/is\_palindrome?text=madam

output: true/false

14. i. Write a PHP script to convert seconds into days, hours, minutes and seconds.   
Sample seconds : 200000  
Expected output : 2 days, 7 hours, 33 minutes and 20 second

ii. Build a Node.js server for the following task,

You are given an integer N. You need to print the series of all prime numbers till N.

Input Format: The first and only line of the input contains a single integer N denoting the number till where you need to find the series of prime number.

Output Format: Print the desired output in single line separated by spaces.

Constraints: 1<=N<=1000

SAMPLE INPUT: 9

SAMPLE OUTPUT: 2 3 5 7

15. Build a SQL database schema for an e-commerce site products master. The master table(s) should fulfil the following criteria

* Each product has a name and unique id
* Each product can have multiple prices, depending on the quantity being bought
* The database will have lacs of records, so give a due consideration to performance

Some sample data for you to use

| Product | Quantity Bought | Unit Price |
| --- | --- | --- |
| Earphones | 1 to 10 | 500 |
|  | 11 to 50 | 485 |
|  | 51 to 200 | 475 |
|  | 201 to 1000 | 450 |
| Memory Card | 1 to 50 | 1000 |
|  | 51 to 200 | 950 |
|  | 201 to 1000 | 900 |
| Hammer | 1 to 5 | 250 |
|  | 6 to 50 | 240 |
|  | 51 to 200 | 230 |
|  | 201 to 1000 | 220 |

Write SQL queries to get the following information

* Name of the most expensive product(s)
* Name of the cheapest product(s)
* Per item price for 75 memory cards
* Per item price for 6 hammer

16. Write a PHP file that can be added to other PHP files using the include or require functions. This file should:

Make a connection to a MySQL database, and log in with valid credentials. The connection resource should be stored in a variable with an appropriate name.

* Create a database 6470 if it does not exist.
* Select the 6470 database.
* Create a table 6470exerciseusers if it does not exist with the following fields:
  + USERNAME VARCHAR(100)
  + PASSWORD\_HASH CHAR(40)
  + PHONE VARCHAR(10)
  + The USERNAME field should be designated as UNIQUE.

If any of these operations cause an error, stop execution and print the error message

Write the registration form Note that all of this part should be done in the same PHP file. The script should respond differently depending on the situation (whether a POST request exists, whether the username is already taken, etc.).

17. Write a PHP file that will output a form containing 3 fields: username, password, and phone number. These fields should be sent via POST to the same file, which should take care of inserting them into a database named 6470 and table named 6470exerciseusers (as shown above), and then confirm the registration by displaying the username and phone number back to the browser.

If the username already exists, your INSERT query should fail if you designated the USERNAME field as unique. You should query the database before attempting the insert, and if the username exists already, display an error message and a blank registration form again.

Write the login form Write a PHP file that will output a form containing 2 fields: username and password. Upon submission of the form, the code should check against the database to see whether the username password pair was correct. If so, display a welcome message. If not, display the message “Invalid username or password” followed by the same login form.

Once again, there should only be one PHP file, and you should redirect to the same place after submitting. The output should be one of three options: 1. The login form. 2. The welcome message, if successful login. 3. The invalid message and the login form, if failed login.

18. Write a HTML form and associated PHP script that allows an administrator to add items to the database of product items. Your script should check for product code duplication before adding an entry. If the code is a duplicate then you should report an error and the conflicting data that already exists.

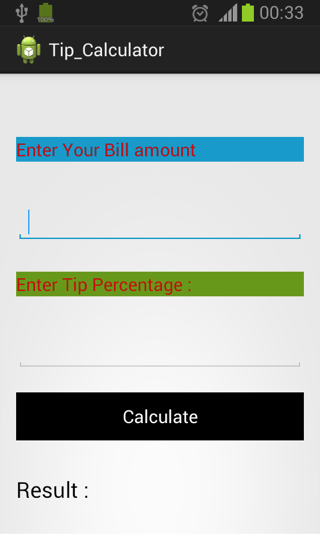
Adding a product should be done via a form that communicates to a PHP script on the server to generate appropriate SQL commands for the MySQL server adding the requested data.

19. Develop the simplest calculator, which takes two numbers and applies addition, subtraction, multiplication and division operations to them. The result is displayed as a complete expression.



20. A tip calculator us use to share tips given to a group of people working together. It is commonly seen in fast food and restaurant related business.

When a bill is given to a customer, a certain percentage of the bill is a tip. The tip can be deducted from the total amount and it will be shred in accordance with the rules guiding tip sharing in the business. Implement this scenario using android.



21. Create the app shown below. Users are initially presented with an "unhappy" character with the corresponding text "I'm so hungry". After hitting the button "EAT COOKIE", the character becomes "happy"

