Basic

* 1. Create database structure for accounting firm. Keep in mind that accounting firm has to keep records of their clients, clients incoming and outgoing invoices, costs and has to keep receipts for at least 5 years. Most frequent query is by client name and yearly income and second one is getting most profitable client  
     **Please see folder – AccountingFirmDB** for create and select test scripts. Files are in sql file format.
  2. How would you find all files  that  ends with “.js”  and bundle them together into one file.

**Solution in Bash command**: (set to project root folder and execute command)   
find . -type f -name '\*.js' -exec cat {} + >> main.js

* 1. You are a forensic computer scientist, and you have to write a program that finds the right person responsible for a murder. the detective sent you an DNK sample of a murderer and list of 3 suspect. Data sent to you:
     1. **DNK**: HHHKLJ140L98IHYYYN **(CODE: HAIR, EYES, RACE)**
     2. **Note for me: You check that it contains all three attributes, then this is the person! Single object – single person, use properties!  
          
        Suspects**:
        1. John Novak
           1. Hair:  Black
           2. Eyes: Green
           3. Race: Asian
        2. Vin Diesel:
           1. Hair: Blonde
           2. Eyes: Brown
           3. Race: Caucasian
        3. Guy Fawkes:
           1. Hair: Black
           2. Eyes: Brown
           3. Race: Hispanic
     3. **Legend**
        1. Eyes:
  2. Black:  140L98
  3. Green: 140A98
  4. Brown: 140A88
  5. Blue:    140L97

1. Hair:
   1. Brown: HHHKLJ
   2. Black:   HHHKLI
   3. Blonde: HHLH1L
   4. White:  HHLH2L
2. Race:
   1. Asian: 1HYYYN
   2. Hispanic: IHYYYN
   3. White: IHYYNN

**Solution is in folder** ForensicComputer. This is Command Prompt/Bash application. Run it command line »php ForensicComputer.php«. Solution found neither of listed persons. I have to have create my own »Perfect Suspect person«, which matched the case.

Practical:

Data: [https://admin.b2b-carmarket.com//test/project](https://admin.b2b-carmarket.com/test/project)

1. Write a parser for provided data, so you get a valid associative array of all the rows
2. Create appropriate database table and write all the rows into it, be careful there could be a lot of rows
3. Generate fake Names and Surnames for all of the Buyers in separated table, one for each buyer.
4. Write php script that prints all of the rows in paginated table
5. Use jQuery to make pagination async, make sure that URL still changes.
6. Write simple search functionality, that searched based on Name,Surname and SaleDate

**Hint**:<https://datatables.net/>, if you decide to use Datatables, make sure that all searching and paging is done in PHP not in JS!  
  
**Solution is in folder »Practical«.**

Firstly configure your's MySQL connection parameters in file "FileParser.php". Before that create MySQL database named "PracticalParseFile" and execute sql script "DataFileTableAndDB.sql"

**FileParser.php**

// exec mysqli connection

$servername = "localhost or IP address of MySQL server";

$username = "root or other MySQL username";

$password = "enter users password";

$dbname = "PracticalParseFile";

Solution is in two php files FileParser.php and DataGrid.php. First file is command line tool, which is run by »php FileParser.php«

Second file is web page using DataTables.Net jQuery component. (you have to wait couple of moments/seconds to fully load)

After running »php FileParser.php«, we have filled data into table stored in MySQL. Please firstly find sql file (DataFileTableAndDB.sql) that is Create table for »datafile« table. Also, you can find testing sql commands »TestData.sql«

For pasing data, there are prepared three CSV files:

* Project.html
* Names.csv and Surname.csv (generated from <http://listofrandomnames.com/>)

For mock up mock up/testing DataTables.Net jQuery component, there is folder named »DataTableTest«