# Database-Driven Web Applications Coursework – Viva Scriptum

**GROUP NAME:** (please overwrite this text with your group name)

## The Basics

1. Please complete the contribution model below – **but please read the other text on this page carefully before doing so!**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kno | Name | Present?  (Write YES or NO) | Contribution (as a %) | |
| K1818372 | Daniel Loten | YES | 20 | % |
| K1845138 | Mars Dmello | YES | 20 | % |
| K1914403 | Faris Ahmed | YES | 20 | % |
| K1902813 | Reem Zemour | YES | 20 | % |
| K1929819 | Catalin Petre | YES | 20 | % |
| Total (should add up to 100% - click on the number and press F9 to recalculate) | | | **100** | **%** |

Please note that the contributions MUST add up to 100%. So, for example, a group of four where all four individuals had contributed equally would involve four contributions of 25%.

Individuals who were not online and did not engage with the preparation of this document should automatically be assigned a zero (0%) contribution unless there are STRONG reasons in the group not to do this. (For example, an individual has been in every workshop and attended every group meeting, and written a substantive number of lines of code, but has internet connectivity issues on the day).

**Ideally, the contribution model should be unanimous i.e. everyone should agree to it.** In the event this cannot be agreed upon, what happens depends on how many people disagree:

**IF TWO OR MORE PEOPLE AGREE ON A SINGLE CONTRIBUTION MODEL:**

* This contribution model should be filled in at the top of this page.
* Those remaining individual(s) who disagree with the contribution model should write IN DISPUTE next to their name above.

**IF THERE IS NO AGREEMENT WHATSOEVER, everyone in the group insists on a unique contribution model and no agreement can be found:**

* Do NOT complete any of the contribution model column at the top of this page.
* ALL members of the group should write IN DISPUTE next to their names.

**ANY INDIVIDUAL WHO HAS “IN DISPUTE” NEXT TO THEIR NAME SHOULD ALSO COMPLETE A CODE OWNERSHIP BREAKDOWN FORM (these are at the end of this document). FAILURE TO DO SO MAY RESULT IN A ZERO MARK BEING AWARDED.**

**If you complete the table above, and there are no IN DISPUTE annotations, no later correspondence will be entered into about individual/group contributions!**

1. Please type the KUNet URL at which your application can be accessed. **Important note: if you application is not accessible on the public internet at a KUNet URL then your group will receive a zero mark!**

|  |
| --- |
| https://kunet.uk/alpaca/controller/frontpage.php |

## Technical foundations of your application

### Data Access

1. Which PHP database APIs have you used in your application? Place an X in the box of all that apply. Please note, **if you leave any out and we find you’ve used them in addition to a preferred option you will be negative marked!**

|  |  |  |
| --- | --- | --- |
| Mysql \* |  | (note, if either of these have been used then your project breaks non-functional requirements and will receive a zero mark) |
| Mysqli |  |
| PDO without OO |  | \* (note, this means the PHP mysqli API – see <https://www.php.net/manual/en/book.mysql.php>. You don’t need to tick this just because your RDBMS is Mysql!) |
| PDO with OO (i.e. using FETCH::CLASS) | x |
| Something else (?!) |  |
| We don’t know |  |

1. How many (PHP) files in your project contain SQL statements? Place an X in **one box only**:

|  |  |
| --- | --- |
| One file and one file only, and it’s solely dedicated to all data access for the app. | x |
| More than one file but it’s done as part of a clearly separated data access strategy – all data access is driven through this group of files, which have no responsibilities other than persistence and data retrieval, and no other SQL statements exist anywhere else.  **WARNING: DON’T CHOOSE THIS IF YOU DON’T UNDERSTAND IT. YOU WILL BE NEGATIVE MARKED! WE’LL CLARIFY THIS IN CLASS JUST TO BE ON THE SAFE SIDE.** |  |
| One file only but it does other stuff as well as data access |  |
| More than one file, and the files do other stuff in addition to data access |  |
| We don’t know |  |

1. Give the filename(s) of the files that involve database access. By “database access”, we mean “includes SQL code” and/or “makes a direct connection to the MySQL database backend”:

|  |
| --- |
| /var/www/html/alpaca/model/dbAccess.php, /var/www/html/alpaca/view/eventadd\_view.php, /var/www/html/alpaca/view/view/cheeseadd\_view.php. |

1. Tell us about your database design and use of SQL:

|  |  |
| --- | --- |
| Our database design and SQL usage are excellent. All entities are separated into appropriate tables. Relationships are used correctly; our database is designed so that duplication is eliminated and it could support future extensions to the application. We are using advanced concepts of SQL that go beyond what was taught in TB1. Database accesses are kept to a minimum. |  |
| We’ve done a good job of designing our database and we use SQL effectively. There are sensible tables that represent the data required for the application. Obvious relationships are present and SQL is used at the level of what has been taught in class. There’s no duplication in our data structure. |  |
| There is a database and we use SQL for simple retrievals and updates. There may be some inefficiencies in both the database design and our SQL. Perhaps there are tables which should have been separated maybe there are no relationships, maybe there are times in our code when we make multiple database accesses when the same information could have been retrieved with a single (more complex) SQL statement. | x |
| We have a database but were only ever able to plug it into our (PHP) code in a very limited sense. |  |
| We don’t feel we know or understand enough to be able to answer the question. |  |

1. Justify your above answer. You can use examples from your code or SQL if needed. You might also consider including UML or an ERD if you have it, or failing that a screen grab from PHPMyAdmin’s visual designer. (Again, the box will expand as needed).  
     
   **IT IS COMPULSORY TO FILL IN THIS BOX. If you leave it blank you will receive a zero mark for the database category of the marking criteria!**

|  |
| --- |
| We have a working database, populated and is able to be called and is interactable with our project. We have selected this level, as we did not include any foreign keys, and the databases don’t interact with each other. |

### Other technical considerations

1. Have you separated your concerns?

|  |  |
| --- | --- |
| Yes, absolutely. We have a full Model/View/Controller design pattern and use Object Orientation correctly. | x |
| Yes, we think so. We’re not quite sure if what we’ve done genuinely is MVC but we’ve separated out the logic/processing from the presentation/user-facing stuff for the most part. And there are some OO classes. |  |
| Probably. For the most part. Most of our code isn’t spaghetti where we mix HTML, PHP and SQL together although there are a few places where this was unavoidable. And we haven’t got any OO classes. |  |
| No. We just concentrated on getting something working. |  |
| We don’t know what that even means. |  |

1. Justify your above answer. Use examples from your code and explain how they work in order to prove to the assessor that your answer to Q4 is correct and that you understand the principles involved (the box will expand as needed). **IT IS COMPULSORY TO FILL IN THIS BOX. If you leave it blank you will receive a zero mark for the technical design category of the marking criteria!**

|  |
| --- |
| Our project has been entirely separated. Everything should run through a controller, no view pages are ever directly linked, you go through the controller and it will display the respective view. We have a separate model section that contains all of the getters and setters for our data, as well as the classes that are used for the database access.  In relation to answer 4, Anything that needs to communicate with the database itself has to be run through the dbAccess file. |

1. Which of the below technologies have you used? (tick all that apply – and notes “used” means “we wrote original code of our own” and not “we pasted in code from the internet” or “we think we used JQuery because the JQuery file is included as part of our project”)

Tick multiple answers if appropriate

|  |  |
| --- | --- |
| Javascript | X |
| JQuery | - |
| AJAX | - |
| Web services | - |
| We don’t know |  |

## Functionality

For each of the functions fill in the box with details that demonstrate its operation (assuming you completed a given function). You should include enough screenshots so the assessor can see the various stages of the function in progress. You may also give excerpts of code where appropriate.

### “Must have” functions

1. **As a casual customer, I need to be able to search for products (cheeses) across relevant categories and fields so that I can find something that I will enjoy eating** – give screenshots and a description of how your application implements the browse and/or search functionality based on these points.

|  |
| --- |
| Our application uses textboxes and forms to allow the user to search our database of cheeses based on separate criteria. Using a POST form method, we have created individual searches that interact with our dbAccess file to retrieve results from the database.    If you go from the homepage to the cheeses page using the menus, the textboxes at the top are there for searching through the database, and to call that information to the webpage. |

1. **As a casual customer, I need to be able to browse the store and add one or more items to a basket without necessarily logging in with an account, so that my shopping experience will not be frustrating** – supply screenshots showing how bookings may be added to the basket, showing the basket updating. Note “without necessarily logging in” – so it should not require an authentication process to add bookings to the basket (although they might have to do so to check out). Also note that a “basket” requires the ability to add **multiple** items at a time before concluding a purchase.

|  |
| --- |
| If you use the top menu’s, navigate to cheese. From there you will be presented with a list of cheeses. If you use the texteboxes on the top you can individually search for cheeses based on your requirements. Using the dbaccess model file separate functions are called based on which textboxes you filled out/which search button you use. For example, our  If you click add to basket, it will be added to your basket. |

1. **As a casual customer, I need to be able to complete the purchase of items stored in my basket so that I may acquire the cheeses I have chosen.** The system should lead the user through supplying the appropriate information required to fulfil their order, and finally inform them that their order has been received.This information should then be stored in a way whereby either a human being or an automated process could realistically have sufficient information to be able to fulfil the order. Supply screenshots and describe the process of how a customer goes from their basket, to supplying their details through to receiving a confirmation that their ticket order has been processed. Additionally, provide details of where/how the order is stored.

|  |
| --- |
| Our system leads the user through all of the relevant fields once they have decided to check out. It shows a list of accepted cards, and takes the customers information. Once these have been filled out, the site updates to show the user that it has been accepted. |

1. **As the shop manager, I need to be able to add new and update existing cheeses.** With screenshots and text, showcase the admin functionality.

|  |
| --- |
| Use the menus at the top to go to cheese. From there, click on the add new cheeses link. You will; be presented with this page.  Input the username: ‘alpaca’ and the password ‘eesoocio’ to log in. From there you will be redirected to our addcheese page. |

### “Should have" functions

1. **As a customer, I need to be notified when the shop makes new products available** Show how a customer is dynamically notified when new cheeses are added (perhaps you could include screenshots before new cheese added/after new cheese added)

|  |
| --- |
| Unfortunately we did not managed to implement this should have. |

1. **As the event coordinator, I need to be able to add events so that cheese enthusiasts can participate in them** – give screenshots and/or a description of how an event can be added.

|  |
| --- |
| To the menus at the top of the page, click events. From there you will find a link to ‘add new events’. Click this and you’ll be taken to the managers log in. Log in with username: ‘alpaca’ password: ‘eesoocio’.  Once input, you’ll be taken to the event add screen. Fill in the relevant boxes and press the ‘eventadd’ button and the event will be uploaded to the database. |

1. **As a deranged cheese-enthusiast, I need to be able to book tickets for events so that I can participate in them.** Use screenshots and text to walk the assessor how this works.

|  |
| --- |
| From the events page you are able to add events to your event basket. However, the add new booking form is currently not accessible through the homepage.. To get access to this page, please follow this link: <https://kunet.uk/alpaca/view/bookingadd_view.php>    This form will interact with the database and store all of the customers details. |

1. **4. As the shop manager, I need to be able to create promotions that automatically appear on the home page.** Walk the assessor through how this function works in your application using screenshots and text.

|  |
| --- |
| ---- unfortunately we did not manage to implement this. |

## Other implementation aspects

1. Is your application responsive?

|  |  |
| --- | --- |
| Yes, fully. We have actively written code so that all device types and screen sizes are supported. |  |
| Mostly. It might be a bit clunky on some screen sizes/devices, it’s not perfect, but we had a go at trying to write code that adapted our apps presentation to be responsive. |  |
| It is usable across different screen sizes/device types but mainly because the presentation is simple enough for this to be the case. We’ve not written any special code in particular that makes it response. | x |
| Our app isn’t responsive. |  |
| Of course our app is responsive. It works really quickly and responds immediately to user input. |  |
| We have no idea what you mean by “responsive”. |  |

If you ticked “yes” or “mostly”, please specify which lines of code (by line number(s) and filename) make your app responsive:

|  |
| --- |
|  |

1. Has accessibility been considered? If so, please explain your answer below:

|  |
| --- |
| To some degree, we included alt text for the visually impaired, which allows the pictures we included to be dictated by certain web browsers/ computer programs. |

## Security

1. (How) does your application distinguish between different users?

|  |
| --- |
| We did this in multiple ways. Our first way of differentiating between users is with the use of the session api. This tracks users on each user session so every individual on the website doesn’t share baskets. It also holds users baskets so if you navigate away, you retain your current basket/items.  The other way we implemented user distinguishing is through the use of a log in portal. There is a log in page that you can access when trying to add cheeses or events. The controller will try to load the respective add page, but if the user has not logged in, it will present a log in portal to be completed before you are able to gain access to the page. It will constantly loop back to the log in page and deny access to the add pages through use of the controller if the correct username and password isn’t included. |

1. Have you considered SQL injection? Explain how your application mitigates against this security risk.

|  |
| --- |
| By limiting the amount of characters able to be input into our text boxes to a low number, we have tried to avoid large strings of harmful SQL to be input into our database. Also, by maintaining the correct data types (example: date type for dates) we can stop harmful and incorrect text being input.  We have also hidden a lot of input forms behind a log in screen, so only the respective authorised employees will be able have access, so your regular user wouldn’t be able to inject SQL into the key parts of our database. |

## Other Things We Should Consider

1. Please give any end-user logins or passwords an assessor might need to explore all the functionality of your application in the box below. Things like admin logins, logins for an existing user (can be helpful rather than expecting the assessor to create new ones) and so on.  
     
   **IF YOU LEAVE THIS BLANK AND THERE’S NO WAY FOR AN END-USER WITHOUT BACK-END ACCESS TO REACH A PARTICULAR PART OF YOUR APP, YOU WON’T GET THE MARKS FOR IT!**

|  |
| --- |
| Username: ‘alpaca’ password: ‘eesoocio’  We used the same log in and password for the database and for our log in portals. Any log in found should use these. |

1. Finally, this is your opportunity to talk about anything that all of the above might not have covered. Is there anything in your application that you think we should pay particular attention to when we’re assessing? Is there a particularly elegant bit of code that does something special that you were really proud of? Is there a particular bit of SQL hackery that you want to show off? Are there design and/or UX components that make your application particularly enjoyable and easy for the end-user? This is your opportunity to explain that your work is awesome, and convince us to give you a stonkingly good mark! 😊

|  |
| --- |
|  |

## Code Ownership Breakdown

The below should only be completed in accordance with the instructions on the first page of this document. If you have all agreed a Contribution Model then there is no need to complete this.

Here is an example:

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name:** | Liam Bates | **KNo**: | K123456 |
| **Filename** | model/dataAccess.php | **Lines** | 1-30, 55-70 |
| **Filename** | view/ticketList\_view.php | **Lines** | 1-134 |

In this example, Liam wrote portions of **dataAccess.php** and the entirety of **ticketList\_view.php** (which was 134 lines in length). Liam has completed the form accordingly. **YOU MUST GIVE SPECIFIC LINES OF CODE AND FILENAMES, OTHERWISE YOUR DOCUMENT WILL NOT BE ACCEPTED, AND IF THERE IS SUBSEQUENT ANY QUERY ABOUT CONTRIBUTION YOUR MARKS BE REDUCED OR EVEN ZERO’ED.**

All students who wrote IN DISPUTE on the first page should complete one of these:

|  |  |  |  |
| --- | --- | --- | --- |
| **Student #1** | | | |
| **Student Name:** | (type name here) | **KNo**: | (type Kno here) |
| Please complete the form below indicating which specific PHP files and which specific lines of code were written by you. You can add additional rows if you need to. | | | |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **If you made any additional contribution to the project which cannot be quantified as lines of code, please describe it below:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student #2** | | | |
| **Student Name:** | (type name here) | **KNo**: | (type Kno here) |
| Please complete the form below indicating which specific PHP files and which specific lines of code were written by you. You can add additional rows if you need to. | | | |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **If you made any additional contribution to the project which cannot be quantified as lines of code, please describe it below:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student #3** | | | |
| **Student Name:** | (type name here) | **KNo**: | (type Kno here) |
| Please complete the form below indicating which specific PHP files and which specific lines of code were written by you. You can add additional rows if you need to. | | | |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **If you made any additional contribution to the project which cannot be quantified as lines of code, please describe it below:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student #4** | | | |
| **Student Name:** | (type name here) | **KNo**: | (type Kno here) |
| Please complete the form below indicating which specific PHP files and which specific lines of code were written by you. You can add additional rows if you need to. | | | |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **If you made any additional contribution to the project which cannot be quantified as lines of code, please describe it below:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student #5** | | | |
| **Student Name:** | (type name here) | **KNo**: | (type Kno here) |
| Please complete the form below indicating which specific PHP files and which specific lines of code were written by you. You can add additional rows if you need to. | | | |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **Filename** |  | **Lines** |  |
| **If you made any additional contribution to the project which cannot be quantified as lines of code, please describe it below:** | | | |