

ASSESSMENT AND INTERNAL VERIFICATION FRONT SHEET (Individual Criteria)

(Note: This version is to be used for an assignment brief issued to students via Classter)

Course Title	B.Sc. (Hons.) Software Development			Lecturer Name & Surname	Thomas Gatt	
Unit Number & Title		ITSFT-506-1612: Server Side Scripting				
Assignment Number, Title / Type		2, Develop a web application for a given scenario / Home				
Date Set		15/12/2021	Deadline Date	15/01/2022		
Student Name			ID Number		Class / Group	

Assessment Criteria	Maximum Mark
<i>KU3.2: Show understanding of the validation techniques available in a framework</i>	5
<i>KU4.1: Show understanding of model commands which can be used to interact with data in a database</i>	5
<i>AA1.5: Examine various methods of debugging</i>	7
<i>AA2.3: Develop a web application using the features of a server side framework</i>	7
<i>AA4.2: Create table associations to be used in an application using the features of a server side framework</i>	7
<i>SE2.4: Develop and design an application programming interface using the features of server side framework</i>	10
<i>SE3.3: Design URLs in a SEO friendly way</i>	10
<i>SE4.3: Modify a web application in order to make it safer using the framework's security features</i>	10
Total Mark	61

Notes to Students:

- This assignment brief has been approved and released by the Internal Verifier through Classter.
- Assessment marks and feedback by the lecturer will be available online via Classter ([Http://mcast.classter.com](http://mcast.classter.com)) following release by the Internal Verifier
- Students submitting their assignment on Moodle/Turnitin will be requested to confirm online the following statements:
 - Student's declaration prior to handing-in of assignment**
 - ❖ I certify that the work submitted for this assignment is my own and that I have read and understood the respective Plagiarism Policy
 - Student's declaration on assessment special arrangements**
 - ❖ I certify that adequate support was given to me during the assignment through the Institute and/or the Inclusive Education Unit.
 - ❖ I declare that I refused the special support offered by the Institute.

Server Side Scripting

Second Year BSc 2021-2022



MCAST

Develop a web application for a given scenario

Assignment 2 - Home

Assignment Guidelines

Read the following instructions carefully before you start the assignment.

If you do not understand any of them, ask your lecturer.

- You are required to use the **CakePHP v4** framework for this assignment. **'Bake' or similar CLI commands are not allowed.**
- **Copying is strictly prohibited and will be penalized** in line with the College's disciplinary procedures.
- The deadline should be specified by your lecturer. When the deadline is due, you shall hand in 2 deliverables:
 - A zip file containing the application, all source code and a database SQL file.
 - You are also required to create a **word document** and **include all the code** from your controllers, models and views in **plain text**. You are to submit this document on the specified platform on the same deadline. Your lecturer will forward you details in order to submit your assignment before the deadline.
- **Marks assigned to each criteria will be determined by the interview performance. i.e. You must be able to answer any questions related to each criterion. You might not be awarded marks associated to a criterion if you fail to answer any questions correctly.**

Introduction

Geek Investments LTD loved your work and would like you to upgrade their system to provide more features. They would like you to create a platform that should allow users to share their investments with other users. They named this concept 'ShareMyTrades'.

Just like any other company, Geek Investments LTD needs this to be developed as fast as possible yet they want it to be secure and scalable. Therefore, you are being requested to make use of CakePHP version 4 as a PHP framework. This not only makes development faster but also easier to maintain.

The users should be able to:

- Manual Registration and Login **and** Login with Facebook or Google
- Add a trade (via the homepage - accessible to Everyone (public) or shared with specific users)
- View trades (homepage)
- Like/unlike a trade
- Access specific trade via direct link
- Delete own trades
- Edit own trades (including trade share settings)

The web app should also cater for two different roles within the system:

- User
 - All the above
 - Admin
 - All the above
 - View all trades
 - View all trades added by a specific user
 - Delete any trade
-

Tasks

Geek Investments LTD have written this user requirement specification document which should be very easy to follow. The web application is divided into various pages.

1. Homepage (Add new Trade AND List trades)

- a. Add new trade (Authenticated Users only)
 - i. Ticker (e.g. Tesla) - should be a dropdown list (required – these can be manually populated in a database table)
 - ii. Notes (optional, minimum of 20 characters)
 - iii. Amount of shares (required, should be numeric and a positive number)
 - iv. Price bought (required, should be numeric and a positive number)
 - v. One image attachment (required, **images only**)
 - vi. Privacy settings: Public/Private (if set to private, a feature should allow the user to share the trade with other users registered)
 - *Public*: This option makes the trade visible to all users (including non-authenticated ones)
 - If the *Private* option is selected, the trade will be available to the owner. This option should allow the user to share it with other users (by using the email address).
 - a. When sharing with other users, the user needs to select from a list the user(s) (by email) that he/she wants to provide access to (*You might want to use the chosen.js client-side plugin for this*).
- b. The homepage should display all the trades that the user has access to:
 - i. Trades set to *Public* (these should be displayed irrespective of the login status).
 - ii. Trades shared with the authenticated user
 - iii. Own trades (trades of the authenticated user)
 - iv. You are expected to show **one whole list**, not separate ones.
- c. Each trade should show the following details:
 - i. Ticker
 - ii. Notes (if any)
 - iii. Amount of shares
 - iv. Price bought
 - v. Image attachment
 - vi. Trade owner (full name)
 - *Admin only*: The full name needs to be linked to a page that shows all trade entries added by the user (see point 2c)
 - vii. Like/Dislike button, a counter of likes and list of users who liked the trade. The dislike button should be shown if the trade is liked and vice-versa. Only authenticated users can like/dislike the trade.

- viii. Date added shown in this format: 20th December 2021 15:20
- ix. 'Edit' button should be visible if the authenticated user is the owner of the trade. The Edit feature should allow the user to change everything (including the image and privacy settings)
- x. 'Delete' button should be visible if the authenticated user is the owner of the trade or an admin. Make sure that only the owner of the trade or an admin can delete the trade (**even if the user uses the direct link**).
- d. Direct link button: The trade should have a unique link using the format: */trade/[ticker-name]/[random-alphanumeric-string]/[user-name-surname]*
 - For example: */trade/tesla/se9e8amd9/thomas-gatt*
 - See the term 'dasherize' on the cookbook to format the user's full name as shown
 - This page, when accessed, should show the same details listed in point 1c
 - This page can **only** be accessed by whoever has access to it (own trade, trade shared with, public trade, admin).

2. List of Users

- a. Authenticated users can view list of users (first name, last name, email) registered with the system
- b. The menu item 'User List' should be visible in the menu if the user is authenticated.
- c. *Admin only*: Clicking on the user should allow admins to view list of trades added by the user.

3. Register/Login & Logout

- a. Manual registration & Login (basic login form and basic registration form)
- b. Implement Firebase's Facebook and/or Google Login for Web
 - i. Save the user's credentials in your table if they registered for the first time
 - ii. Otherwise, retrieve data from the table and authenticate the user
- c. Logout (should be visible if the user is logged in only)
- d. You should at least store the following data:
 - i. First Name
 - ii. Last Name
 - iii. Email

4. My Trades / User Trades

- a. 'My Trades' menu item should be visible if the user is logged in
- b. It should contain a list of trade entries (ticker and amount of shares) added to the website by the authenticated user. This list should include all trade entries and each trade should be linked as described in point 1d.

- c. *Admin only*: View list of trades added by a user – (as requested in point 2c)

5. API

Geek Investments LTD would also like you to provide easy access to the logic you've created in your application so that others can make use of this data. You should make use of CakePHP's REST function to expose your API via REST.

- a) Get list of trades and their likes added by a **specific user** (GET)
- b) Delete a **specific** trade (DELETE)
- c) **Add** a new trade including attachment (POST)

6. Debugging

While CakePHP's core Configure Class settings can really help you see what's happening under the hood, there are certain times that you'll need to log data to the disk in order to find out what's going on.

Make use of CakePHP's Log class to log specific events related to trades and its likes/dislikes. The logs should include both successful and failed actions. By creating **specific log scopes**, make use of the correct logging levels and log the following events:

1. New trade
2. Like/Dislike a trade
3. Logins
4. Logout

Details to log

- Date & Time
- User (if logged in)
- IP Address
- Message
- User/Trade ID if applicable
- Any other information you believe is important to log

Design

You are expected to present a simple yet neat design. Feel free to use Bootstrap. Please **refrain** from using the `<table>` tag to display lists. Use cards or lists instead.

--End of Assignment--

Assignment Rubric

Criteria	Description	Marks	Awarded
AA1.5	Examine various methods of debugging		
	Log Scope for the TradesController set up	2	
	Correct use of logging levels used in Log Scope	2	
	Correct use of general debug logging throughout application	2	
	List of users page implemented as described	1	
AA2.3	Develop a web application using the features of a server side framework		
	'New trade' implemented correctly and as described	3	
	Correct privacy settings set for a Trade	2	
	'Dislike' trade correctly implemented	2	
SE2.4	Develop and design an application programming interface using the features of server side framework.		
	Implement a REST API method to get list of trades and their likes for a specific user	3	
	Implement a REST API method to delete a specific trade	3	
	Implement a REST API method to add a new trade (including attachment)	4	

Criteria	Description	Marks	Awarded
KU3.2	Show understanding of the validation techniques available in a framework		
	Correct use of <i>Validator</i> Object as described	0.5	
	Ability to Delete a trade as described	1	
	Ability to Edit a trade as described	1.5	
	A trade can be liked	2	
SE3.3	Design URLs in a SEO friendly way		
	Correct use of the <i>Router</i> & <i>Inflector</i> class for accessing a specific trade	2	
	'My Trades' implemented as described	2	
	Admin actions implemented as described	2	
	Trade shared with other users as described	4	
KU4.1	Show understanding of model commands which can be used to interact with data in a database		
	'New trade' – image uploaded, checked for extension and stored correctly	2	
	Homepage implemented as described	3	
AA4.2	Create table associations to be used in an application using the features of a server side framework		
	Correct association types used in Users and Trades controllers (1 mark each)	2	
	Correct use of <i>contain()</i> ORM query function	1	
	Trades in the homepage shown as described and according to access levels. Users are not allowed to access a specific trade via the direct link if they have no access to it.	4	
SE4.3	Modify a web application in order to make it safer using the framework's security features		
	Firestore Google/Facebook Login implemented correctly	5	
	Correct use of the <i>Authenticator</i> component	1	
	List of users (and a counter) who like the trade correctly shown. Dislike button only shown if the trade is liked and vice-versa	4	

Total Mark (out of 61)	
Comments	
