Advanced Higher Computing Science

Web design and development project workshop materials

This edition: December 2024 (version 1.0)

Introduction

This document is for teachers and lecturers and/or Advanced Higher Computing Science candidates.

This document contains workshop activities originally devised for an Understanding Standards event held in 2023. The workshop activities focused on the problem description, requirements specification, design, implementation and evaluation of an example Advanced Higher web design and development project.

Workshop activity 1: problem descriptions

Read the following problem description:

Example 1

This project will involve current basketball teams in the NBA and the top 50 rated players. It will use **database design and development** in conjunction with **web development** to display data and output to the user. The search pages can only be accessed by registered users of the website. To search for players and teams, users are required to register or login using their stored username and password. They can then enter a player of their choice. The player details will then be displayed with details such as player name, team, position, height, points per game, number of championships and number of MVPs. Now, the user can rate and comment on the player giving a rating out of five stars. Users can also enter a team, and the selected team details will be displayed with details of all the team's players.

Mandatory requirements

- Forms will be used for users to enter their username and password.
- Forms will be used to store the user's ratings and reviews of the players.
- External CSS will be used for each page of the website.
- Media queries will be used to create two different layouts for different screen sizes.
- Session variables will be used.
- Assign variables with user inputs.
- Process the form data and display the user's input in tables.

Things to consider:

- Is the focus of this project a database development that integrates with a website, or is it a web development that integrates with a database?
- Advanced Higher projects must validate all inputs. Consider whether this requirement has been considered in example 1.
- Consider whether the intended use of the web requirements mentioned in example 1 meets the requirements of an Advanced Higher project.
- Consider whether the intended use of a database meets the requirements of an Advanced Higher project.

Example 2

Example 2 meets the standard required for the Advanced Higher problem description.

This project is concerned with the creation of a website about NBA basketball teams and players. The website will be appropriate for AH level since it will:

- use HTML form elements to enable users to enter and validate their username and password
- use external CSS to format each page of the website to ensure that the layout of each page is consistent
- make use of a media query to generate layouts that depend on the screen size being used
- use a session variable to store the username entered so that it can be used on subsequent search pages visited by the user to provide a personalised message
- assign values entered in HTML forms to PHP variables
- use PHP to process any data submitted by the user
- validate all input to HTML forms and generate appropriate error messages when invalid values are entered

The website will integrate with a database. This database will be used to:

- store details of NBA players, NBA basketball teams, registered users and reviews in separate tables
- allow users to register a new account
- allow registered users to login and access the search pages and review players

Workshop activity 2: requirements specification

Example 3

Example 3 goes beyond the standard required for the Advanced Higher problem description.

This project is concerned with the creation of a website about NBA basketball teams and players. The website will be appropriate for AH level since it will:

- use HTML form elements to enable users to enter their username and password
- use external CSS to format each page of the website to ensure that the layout of each page is consistent
- make use of a media query to generate layouts that depend on the screen size being used
- use a session variable to store the username entered so that it can be used on subsequent search pages visited by the user to provide a personalised message
- assign values entered in HTML forms to PHP variables
- use PHP to process any data submitted by the user
- validate all input to HTML forms and generate appropriate error messages when invalid values are entered

The website will integrate with a database. This database will be used to:

- store details of NBA players (player ID, name, team, position, height(m), points per game, number of championships and number of MVPs)
- store details of NBA basketball teams (team name, state)
- store details of registered users of the website (username and password)
- store details of player reviews submitted by registered users (review ID, username, player ID, rating, review, date)
- allow users to register a new account
- allow registered users to login and access the search pages and review players
- allow only registered users to search for a player and display the player's details together with all reviews of that player that have already been submitted
- allow only registered users to search for a team and display details of its players
- allow only registered users to review a player and submit their review

Completed requirements specification for example 3

Since example 3 exceeds the requirements of the Advanced Higher problem description, the number of end-user and functional requirements below also exceeds the Advanced Higher project requirements.

End-user requirements

Requirement number	The end users of the solution should be able to:
EU 1	Enter username and password to register
EU 2	Login using their username and password
EU 3	View personalised messages on search pages
EU 4	Search for details of specific NBA player
EU 5	View details of required player together with reviews previously submitted
EU 6	Review player by providing a rating and adding a comment
EU 7	Search for details of a specific team
EU 8	View details of players of required team
EU 9	Navigate easily between pages of the site

Functional requirements

Requirement number	The solution is required to:
FR 1	Store details of NBA players, NBA teams, NBA player reviews and website users in four related database tables
FR 2	Connect to database to execute SQL queries
FR 3	Use HTML registration form to enter username and password
FR 4	Validate username entered has between 3 and 10 characters and password has at least 12 characters
FR 5	Execute SQL query to add user's details to the database
FR 6	Use HTML login form to enter username and password
FR 7	Validate username and password entered
FR 8	Execute SQL query to search for user login details to check that correct password has been entered
FR 9	Only display search pages once a user has registered or logged in
FR 10	Display personalised message on each search page
FR 11	Use HTML player search form to enter name of NBA player required
FR 12	Validate player name entered — this will be a required value

Requirement number	The solution is required to:
FR 13	Execute SQL query to search for player using player name entered by the user and display the player's details together with previously submitted reviews
FR 14	Use HTML review form to submit review of NBA player
FR 15	Validate values entered — all values will be required
FR 16	Execute SQL query to add user's review to the database
FR 17	Use HTML team search form to enter name of specific NBA team required
FR 18	Validate player team entered — this will be a required value
FR 19	Execute SQL query to search for team using the team name entered by the user and display the team's details together with details of that team's players
FR 20	Assign all values submitted from HTML forms to PHP variables
FR 21	Use a media query to generate different layouts that depend on the size of the screen being used — one layout for screens with less than 600 pixels, one layout for screens with 600 or more pixels
FR 22	Use session variable to store username entered at registration or login so it can be used on subsequent pages visited
FR 23	Use external CSS to ensure a consistent layout on each page of the website
FR 24	Validate all inputs to HTML forms and provide error messages for the user
FR 25	Pages of the website should be easy to navigate

The requirements specification forms the 'golden threads' that run through the project development.

Consider the importance of having clearly defined end-user and functional requirements in the requirements specification:

- when creating the project plan
- at the design stage of the development
- during the implementation of the solution
- when creating the final test plan
- at the evaluation stage of the development

Workshop activity 3: discussion points

Design

Consider the design tasks that must be completed for this project.

- How many pages should be indicated on the site navigation structure? Discuss how the site navigation structure could be used to indicate the intended use of the username session variable.
- What processes will be executed by the PHP page that will receive data submitted from the player search form?
- How much detail must be indicated on the entity-relationship diagram to show the relationships that will exist between the entities in this system?
- What details must be indicated on the user interface design of the HTML page used to generate the player search form and the PHP page used to display the search results?

Complete list of design tasks

Design task 1: design of Advanced Higher concepts

- create pseudocode to show the assignment of variables and the processing of form data
- create site navigation structure
- indicate intended use that will be made of session variables
- design all input validation that will be necessary
- show the intended effect of media query or queries

Design task 2: design of integration

- create a data dictionary for the tables that will be used to store the player, team, user and review tables
- create an ERD to show the relationship between these tables
- design the connection to the database (if not already indicated in the pseudocode showing processing of form data)
- design each query required for the solution

Design task 3: user-interface design

- complete the user-interface design for each input screen, showing input validation and underlying processes behind any buttons or menu options
- complete the user-interface design for each output screen

Design task 4: design matches requirements

- check that a design that meets all requirements listed in the requirements specification has been submitted

Implementation

Consider the solution being developed.

Implementation of integration

- What evidence should be included to demonstrate that all of the implemented queries work correctly?

Implementation of user-interface

- What evidence should this candidate include to gain full marks?

Log of ongoing testing

The following log was submitted.

- Does this log provide sufficient evidence of the ongoing testing that would be needed when the solution is being implemented?
- What evidence that has been omitted should be included?

What is being tested?	Issue encountered	How it was solved	Resource used
Connection to database	Instead of running PHP code, it printed out the code onto the screen.	I changed the URL of the webpage to look at localhost, the address of the server so that it could interpret the code.	Class notes
Connecting to the database	It would not connect as it stated that the database being used wasn't accessible.	I had put the wrong database name in and once it was changed, the connection was then successful.	Class notes
Whether data entered in the registration page was passed to PHP variables	The server wasn't accepting the data that was entered in the form.	The names of the form variables were not the same as the ones I used on the PHP variable.	https://sebhastian.com/php-form-handling/?utm_content=cmp-true
Checking that the registration page would insert into the database	The database had a new record, but the fields had the wrong values.	The variables used in the SQL statement were round the wrong way. Once I had sorted them the right way, the new record was correct.	Stackoverflow
Ability to login and have personalised message displayed on player search screen	New users could register but no personalised message was shown.	The session variable for username wasn't being passed to the player search screen because the session hadn't been started on the search screen page. Once this was added, the username and welcome message were displayed.	https://stackoverflow.com/questions/17242346/php_session_lost_after_redirect
Testing to make sure that the search for player page worked	The search returned no results even though the player name had been typed correctly.	The SQL statement was incorrect as the PHP variable used to search for the player was wrong. Once that was changed, the correct player details were shown.	Stackoverflow
Testing to make sure media query works when pages displayed on a small screen	Query didn't work, but the large screen layout was still shown.	The code in the media query for small screens was incorrect. Once I fixed the error, the media query worked as I wanted.	https://www.w3schools.com/css/css3_mediaqueries.asp and https://www.w3schools.com/css/css3_mediaqueries_ex.asp

Evaluation: evaluation of maintainability and robustness

Consider the solution being developed.

- Advanced Higher candidates are expected to refer to specific types of future maintenance. Consider features that could aid or hinder such maintenance.
- Consider whether it would be appropriate to be critical of the limited input validation that had been included in the solution.