

Dynamic Data-driven Design

Introduction

The High School Art Gallery prototype that you've examined in the last few activities already has some nice functionality, but it can still be improved in many significant ways. This is your chance to put to use your knowledge of design, data abstraction, and algorithms to create a unique computational artifact.

How would you improve the site? How would you change the appearance, navigation, and information provided? What features would you add or modify?

Materials

- Computer with
 - Browser
 - Cloud9 workspace
- Brainstorming Tools
 - Sticky notes
 - Scrum Poker Cards or a Scrum Poker App on a device (1 per student)

Resources

[CSP Presentation Rubrics](#)

Procedure

Part I: Planning

1. Form teams of four to six as directed by your teacher.
2. Meet or greet each other to practice professional skills.
3. Review the rubric for this project as directed by your instructor. The goal of this project may be either:
 - Project 2.2.4 – Extend the capabilities of the High School Art Gallery website.
 - Problem 2.2.4 – A client has seen your work on the High School Art Gallery and wants you to create a solution for their business. Repurpose the code to a new business context of your own imagining.
4. Once your team has assembled, professionally greeted each other, and determined whether you will be expanding the High school art gallery or working on a client's project, your team

should go through a formal brainstorming process as directed by your teacher before continuing to the next step.

5. Plan your workflow and document the planning.

This includes:

- Setting team norms for assigning tasks
 - Maintaining version control
 - Resolving problems
 - Balancing workload
 - Ensuring that everyone is contributing
 - Documenting the project
6. Consider using a Sprint Task List and Product Backlog to help you document your work on the project. This may make it easier at the end of the project to describe the work that you've done and your intended next steps in both your presentation and your reflection.
7. Use Scrum Poker to help you determine which tasks you should prioritize in the upcoming sprint.
8. As a team produce a short sales pitch explaining the improvements you plan to make and why you see them as important. Document this as it will be helpful when you present at the end of the project.
9. Once you have distributed tasks amongst your team, record your plan for completing your task in your engineering notebook. Use written descriptions, diagrams, and pseudocode. Be sure to include the languages you will use and what role they will play in your technical solution.

Part II: Development and Evaluation

10. Begin implementing your solutions.
11. If you are modifying the existing High School Art Gallery project, your instructor may direct you to recreate the database structure in your own tables rather than using their database. It consists of the two tables shown below:

```
mysql> DESCRIBE artists;
```

Field	Type	Null	Key	Default	Extra
username	varchar(128)	YES		NULL	
password	varchar(128)	YES		NULL	
firstname	varchar(128)	YES		NULL	
lastname	varchar(128)	YES		NULL	

```
mysql> DESCRIBE images;
```

Field	Type	Null	Key	Default	Extra
username	varchar(128)	YES		NULL	
filename	varchar(128)	YES		NULL	

thumbnail	varchar(128)	YES		NULL	
-----------	--------------	-----	--	------	--

12. Document your work!
13. Keep a daily journal entry of each day's work.
14. Capture and save screenshots of the work in progress.
15. As a team conduct stand-up Scrum meetings to assess the progress of all of the tasks.
16. Groom the product backlog and daily Scrum task lists.
17. Repeat as necessary steps 9 through 14 until the project is finished.

Note: If you attempt to upload large files (over 3MB), you may encounter unexpected results.

Part III: Presentation

18. Prepare a four to six minute presentation of your work as a team. Explain the improvements to the site that you proposed (sales pitch), the work that you did, any outstanding accomplishments, and any challenges. Explain the next steps your team would take in continuing to work on this project. Describe the impact that collaboration had on your work. Include a digital slide show with screenshots that highlight your improvements to the site. Remember to have each team member actively involved in the presentation.
19. Present your project in front of the class using your best professional and interpersonal skills as directed by your instructor.
20. While you are in the audience, be an active listener. Write an evaluation of the usability of the website to give to the presenting team. Evaluate based on the following usability criteria:
 - Structure: The interface should be organized, with related elements together.
 - Simple: Common tasks should be easy.
 - Visible: Information and options should be easy to find without the distraction of unnecessary information.
 - Feedback: User should be informed of actions, changes in state, and errors.
 - Tolerance: Mistakes should be easy to undo, and reasonable input should be interpreted.
 - Reuse: Design should be consistent across components.

Conclusion Questions

1. Describe the Internet protocols and services your project used.
2. Practice Opportunity for the *Create* Performance Task

Respond to two or more of the following prompts as directed by your instructor.

- "Create a 50-59 second video in which you demonstrate the running of at least one significant feature of your program." (Adapted from College Board Create Performance Task Part 1.)
- "Identify the purpose of your program and explain what the video illustrates. (Approximately 150 words)" (Adapted from Create Performance Task Part 2a.)
- "Describe the incremental and iterative development process you used, focusing on two

distinct points in that process. Describe the difficulties and/or opportunities you encountered and how they were resolved or incorporated.” (Adapted from Create Performance Task Part 2b.)

- “Describe how each algorithm within your algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program. (Approximately 200 words)” (Adapted from College Board Create Performance Task Part 2c.)
- Explain how abstraction helped manage the complexity of your program. (Approximately 200 words) (Adapted from College Board Create Performance Task Part 2d.)
- Paste your previous four responses into a new document and clearly label them 2a through 2d. Use screenshots to capture your entire program code, and paste the image(s) into the document, labeling the images 2e. Draw an oval around the segment of code corresponding to the algorithm of algorithms you described in deliverable (c). Draw a rectangle around the segment of code corresponding to the abstraction you described in deliverable (d). Save the document as a PDF. (Adapted from College Board Create Performance Task Part 2e.)

Note

This last set of deliverables is adapted from the official College Board Create Performance Task but they do not duplicate the content of College Board Task or Rubric. The task provided here contains elements that are different than the College Board Performance Task and Rubric. Please reference official College Board materials.

3. Reflect on the following aspects of the collaborative process.
 - - Division of workload
 - - Significant contribution that you personally made while developing code
 - - Significant contribution that your partner made while developing code
 - - Important question or valuable piece of feedback that you directed to your partner
 - - Important question or valuable piece of feedback that that your partner directed at you
4. What went well in the collaborative process and what could have been improved?
5. What did you personally do well during this project, and what could you work to improve in the future?
6. What would you do next if you were to continue to work on this project?

Project 2.2.4 Dynamic Data Driven Design Rubric

	4	3	2	1
MySQL	MySQL queries are executed in an inefficient manner (not using more than needed).	MySQL queries are mostly executed in an efficient manner (not using more than needed).	MySQL queries are executed in a somewhat inefficient manner (using more than needed).	MySQL queries are executed in an inefficient manner (using more than needed).
Effectiveness	Code accomplishes the tasks set forth in the sales pitch. Task implementations are correct and efficient .	Code accomplishes most of the tasks set forth in the sales pitch. Task implementations are mostly correct and efficient .	Code accomplishes some of the tasks set forth in the sales pitch. Task implementations are somewhat correct and efficient .	Code accomplishes none of the tasks set forth in the sales pitch. Task implementations are not correct or efficient .
CSS	Code incorporates CSS modifications to significantly improve the appearance of the site.	Code incorporates CSS modifications to improve the appearance of the site.	Code incorporates CSS modifications that may not improve the appearance of the site.	Code does not incorporate CSS modifications.
JavaScript™	Code incorporates JavaScript in a new or novel way to improve the functionality of the website.	Code incorporates JavaScript that does improve the functionality of the website.	Code incorporates JavaScript that does not improve the functionality of the website.	Code does not incorporate JavaScript.
Content vs. Style	HTML is never used to add style.	HTML is infrequently used to add style.	HTML is somewhat frequently used to add style.	HTML is used frequently to add style.
Client-side vs. Server-side	Code uses server-side scripting and client-side scripting where appropriate.	Code mostly uses server-side scripting and client-side scripting where appropriate.	Code mostly fails to use server-side scripting and client-side scripting where appropriate.	Code fails to use server-side scripting and client-side scripting where appropriate.
Other comments:				