Final Project FE Development

Total Points Possible: 100
Is a PASS/FAIL grading system
Passing Grade: 80/100

Sample images of what a Final Project would look like:
http://spring2023.uoadev.com/FinalSiteExample.pdf

Not Allowed:
- Using a website generator or template is not permitted
- Copying another student's code and presenting it as your own

Requirements:
- Site must be hosted on http://spring2023.uoadev.com
- The final project needs to be completed by 5pm on Nov 10th
- Mark and Jordan can review your site at any point before 5pm on Nov 10th to let you know if you have enough points to pass the final
- You must complete a slide that contains details about yourself and includes a link to either the final project or a project you want to showcase to employers. This will be shared with the University of Arkansas and sent to employers in the northwest Arkansas area. A sample can be found here
    - http://spring2023.uoadev.com/FinalSlideExamples.pdf

Final Presentation (Required):
- Give a 5-10 minute presentation on your webpage. Presentations will be on Nov 14th and your presentation should include
    - Showing each page of your site
    - Explain one challenge you faced while building your site and how you overcame it
    - Explain one part of your site that you had the most fun building
- Presentations will be in alphabetical order of the class roster
- If you cannot attend the final please make arrangements to present either before or after Nov 14th. If you do not present your final project you will not pass the class

Component	Requirements	Details	Concepts	Points
CSS	- All pages must use the same **external** central CSS - Style must be cohesive and fits your content.	- Pages may make local modifications to the CSS of that page - Must style at least: - Textbox - Buttons - Paragraph	CSS	10

Header or Nav Bar	● All pages (login page optional) must contain the same header which includes a menu bar.	● Menu bar allows navigation to different parts of the site without breaking the site	● Links ● Navigation	10
Footer	● Has "company" information ● Has links to social website ● Remains at bottom of screen.	● Footer ● Links - Social links do not have to link to an actual company social site but should link to social site (Example facebook, instagram, linkedin, etc)	● Footer ● Site Design	5
Secure Page	● A portion of the site must be secured via a username and password ● User must be able to log in ● User must be able to logout and no longer access secure portion	● Username of Letters no spaces no numbers; ● Password can be any character no spaces ● Cannot bypass via normal user methods (clicking, changing the URL) ● Must display an error message if spaces or numbers are entered into either the username or password ● On successful login the page should redirect to a new page to indicate they have logged in successfully	● Forms ● Form Validation ● Session Objects ● Regex ● Javascript	10
Main Landing Page	● Welcome page. ● Has links to actions user can take	● Site must have at least **6 pages total**	● HTML ● Links	5
ADD TO STORAGE	● Add information to a table or database	● Design your table (or storage object) ● Get info from User	● Storage	10

	● May use any storage method but must be persistent	● Add to storage		
READ FROM STORAGE	● Read information from a table or database ● May use any storage method but must be persistent	● Read information from your storage. ● Display to user	● Storage	10
UPDATE STORAGE	● Update row in a table or database ● May use any storage method but must be persistent	● Get info from a user ● Update info in a table	● Storage	10
Number Guessing Game	● User must guess a random number between 1 and 100. ● User must place a "bet" to play. ● Bet must affect the user balance depending on win or lose	● Give user a starting balance. ● Create a table (or use other storage) ● Store balance in table or database ● User must have that much money in their account in order for bet to be valid. ● If user can guess in set number of guesses, wins money, added to account, balance shown ● If user does not guess correctly, money should be subtracted from account and an updated balance shown. ● "Balance" must be recovered on next login ● Must use at least 1 function call	● If statements ● Loops ● Storage	15

		● User should only be able to access Number Guessing Game Page after a success login on your secure page		
Bank Page	● User must be able to access a page to see their main balance and be able to withdraw or deposit funds into their account	● User must be able to see an account balance ● User must be able to deposit funds to their account balance ● User must be able to withdraw funds to their account balance ● Balance must be stored in persistent storage ● User should not be able to enter a negative number into the deposit or withdraw amount ● User should only be able to enter numbers (no letters) into the deposit or withdraw amount ● User should only be able to access Bank Page after a success login on your secure page	● Error handling ● Storage ● If statements	15