SOFTWARE SYSTEM DEVELOPMENT - Monsoon – 2024
Assignment 2 – HTML, CSS, JavaScript
Submission Due Date: 5 October 2024, 08:30 PM

IMPORTANT INSTRUCTIONS

1. This assignment is an individual submission, **NOT** a group activity.
2. Total Marks of the assignment is 150 with duration of 3 weeks.
3. Inputs/output should fit the criteria mentioned in respective questions. Unless it is specified, all input/output criteria are open to interpretation.
4. Evaluation will be conducted based on a fixed grading rubric (syntax, logic, input and output) and the marks are divided as per prescribed weightage in respective question.
5. For queries, please share your queries via Moodle.
6. Please follow the submission instructions carefully.
7. Submissions should be done via Moodle only.

Submission Guidelines:

- Please create one ZIP as .ZIP
- This ZIP should include three folders Q1, Q2, Q3, Q4, Q5, Q6
- Each of these folders should contain sub-folders HTML (with html files), CSS (in case of external css files), JS (in case of external JS files), SQL Files and README.txt (if you wish to include any instructions for evaluator, please do here.) You may leave these folders blank if your submission doesn't use any external CSS/JS files.

Q1: Import data from empdetails.csv file into a table called "Person" with EmpID as primarykey in your local mysql database. Use Source Files: empdetails.csv (https://github.com/vranonymous/data/blob/main/empdetails.csv) to answer this question. Write the following stored procedure(s) using SQL to perform tasks as listed below: (Total: 40 Marks)

a. Create a table called "hike2024" with columns **{HikePK, EmpIDFK, FirstName, LastName, Gender, WeightInKg, LastHike, LastSalary, NewHike, NewSalary}** with respective reasonable datatypes. Populate NewHike, NewSalary columns using your stored procedure with employee information whose weight is less than 55kg only. NewHike column should be populated with 12% increase to LastHike and recalculate the NewSalary using LastSalary column based on the hike percentage. Re-running the stored procedure should delete the data from existing hike2024 table and should repopulate the hike2024 table. **(10 Marks)**

Example: If Sai Anirudh's weight is 53kg, LastHike is 10% and LastSalary is 300000, then the NewHike is 22% and New Salary is 366000.

b. Create a table called "**PersonJoining**" with columns **{PJoinPK, EmpIDFK, FirstName, LastName, DateofBirth, Age, DateofJoining, DayofJoining, MonthofJoining, YearofJoining, WorkExpinDays}** with respective reasonable datatypes. Populate DayofJoining, MonthofJoining, YearofJoining, WorkExp columns using your stored procedure with employee information. Re-running the stored procedure should delete the data from existing PersonJoining table and should repopulate the PersonJoining table. **(10 Marks)**

Example: If Sai Anirudh's DateofJoining is 10-10-2011, then DayofJoining is 10, MonthofJoining is October, YearofJoining is 2011 and WorkExp is 3945 days.

c. Create a table called "PersonTransfer" with columns **{PTPK, EmpIDFK, FirstName, LastName, Gender, DateofJoining, CurrentRegion, NewRegion}** with respective

reasonable datatypes. Populate NewRegion column to “DC” using your stored procedure for employees with Gender “F” whose Work Experience is more than 10 years. Populate the NewRegion column to “Capitol” using the same stored procedure for employees with Gender “M” whose Work Experience is more than 20 years. Re-running the stored procedure should delete the data from existing PersonTransfer table and should repopulate the PersonTransfer table. **(10 Marks)**

Example: If Sai Anirudh’s Gender is “M” and his Work Experience is more than 20 years, update his NewRegion entry to “Capitol”.

d. By using the Person table, write a SQL query to display data in the following format. **(10 Marks)**

EmployeeRegion	**No. Of Employee born between 00:00 hours to 09:00 hours**	**No. Of Employee born between 09.01 hours to 16.00 hours**	**No. Of Employees born after 16:01 hours until 22:59 hours**
Region_1	*Values*	*Values*	*Values*
Region_2	*Values*	*Values*	*Values*

Q2: Implement a Pachisi board game using any 3D JavaScript Library of your choice. *(Example: three.js, Aframe.js, D3.js, Babylon.js, zdog,js, Cannon.js, PlayCanvas, LightGL, Phoria.js, JQuery (3D), Tilt.js etc.)* **(30 Marks)**

Source:

- https://www.youtube.com/watch?v=nbwrtCd67D8
- https://winning-moves.com/images/classicParcheesirules.pdf

Q3: Capture and Log Mouse Events in a file for above Pachisi Game **(20 Marks)**
Hit: MouseEvents DOM - https://developer.mozilla.org/en-US/docs/Web/API/MouseEvent

Q4: Create a MERN Stack Shopping Cart Website excluding Payment Gateway. All properties of the website are open to your interpretation and assumption. (For Example: User Management, Product Catalog, Shopping Cart, Order Management, Cart View) **(50 Marks)**

Q5: For above created SHOPPING CART website. populate the product catalogue articles and advertisement images using Promise Calls externally through an external JSON file. The layout of the page, design and positioning of the product articles and advertisement is your choice. You may implement this outside the MERN stack setup. **(10 Marks)**