**Test Table Cartoon Game**

|  |  |  |  |
| --- | --- | --- | --- |
| **Description** | **Did it work? Y/N** | **What did i do to fix it?** | **Works now?**  **Y/N** |
| Creating multiple forms | Y | None needed | - |
| Ensured all forms opened correctly | N | Used Show method to ensure that all forms open as they should | Y |
| Ensured no forms caused errors | N | Passed the current User onto every form when opening | Y |
| Switching between forms | N | Added a Show method and used a back button to ensure the user can go back to the form before if they made a mistake | Y |
| Verified buttons opened the correct form | Y | None needed | - |
| Create database for login | Y | None needed | - |
| Login database connected without issues | N | Fixed connection string and ensured database exists | Y |
| Create database for cartoons | Y | None needed | - |
| Cartoon database connected without issues | N | Added a Cartoon class to define a cartoon object with ID, Name, Hints, and Image Path.  And a Cartoon Database class that handles database connection and retrieval of cartoon data. | Y |
| Log in form  loads correctly | Y | None needed | - |
| Users can log in  And it's being saved | N | Added:  this. Table Table Adapter. Fill (this. database1DataSet.Table);  And  This. Table Adapter Manager. Update All (this. database1DataSet);  To  update and fill the dataset and save into the database | Y |
| Shows error for invalid login | N | Added validation by adding an error message box on incorrect login | Y |
| Password is  hashed | N | Used SHA256 hashing before storing passwords | Y |
| Password is  hidden on log in form | N | Set textbox Password. Password Char = '•'; - to make it invisible | Y |
| Register form loads correctly | Y | None needed | - |
| New users can register | N | Added validation and SQL Insert query by saying that if the user already exists, they should log in | Y |
| Password is hidden on register form | N | Set textbox Password. Password Char = '•'; - to make it invisible | Y |
| First cartoon appears correctly | N | Fixed image path issue and the database query in the Load Cartoons and Display Cartoons | Y |
| User can enter an answer and submit | N | Added a text box and a submit button to make sure the user can answer and submit any cartoons | Y |
| Incorrect answers show another hint | N | Added an if else statement to go through the hint display, they change based on the guess count | Y |
| Show the actual answers after 3 wrong guesses | N | Modified the submit button method to display the correct answer after 3 incorrect attempts with an else statement | Y |
| Skip button appears after 3 wrong answers | N | Set pictureBox1.Visible = true; to make it visible so the user can skip a cartoon | Y |
| Next cartoon loads after 1 second after correct answer | N | Used Task. Delay (1000) to delay the next cartoon | Y |
| Shows a message when all cartoons are completed | N | Added a check in Display Cartoon to show a message when all cartoons are finished | Y |
| Save answers to the correct CSV file | N | Modified the path used to save the CSV file to ensure its correctly located in the project folder | Y |
| Answers are displayed at the end for the user in the right format (Answer, Result, Score) | N | Modified the Show User Answers to properly read from the CSV file and display the answers with results and scores in the correct format | Y |
| CSV format is done correctly (Username, Answer, Result) | N | Ensured that the Show Answer to CSV is written with the correct format: Username, Answer, and Result | Y |
| Cartoon images load into the  picture box correctly | N | Added an SQL query to SELECT \* FROM Cartoons | Y |