|  |
| --- |
| Neelima Saraf |
| CSharp Test Series |
| Multiple choice questions test for C# learners. |

|  |
| --- |
| Neelima Saraf  1/5/2022 |

Table of Contents

[**Product vision** 3](#_Toc92373842)

[**Project overview** 4](#_Toc92373843)

[**Design decisions** 5](#_Toc92373844)

[**Database design** 6](#_Toc92373845)

[**Website layout:** 7](#_Toc92373846)

[**Review Questions:** 11](#_Toc92373847)

[**Operations documentation** 12](#_Toc92373848)

[**Requirements documents** 13](#_Toc92373849)

[**Support documentation** 14](#_Toc92373850)

[**System documentation** 15](#_Toc92373851)

[**User documentation** 16](#_Toc92373852)

**Product vision**

This is a online assessments tool for C# learners. Users can access their c# skills by taking an online test base for a specific topic and get the test result upon completion of the test.

This product is useful for:

* Students
* Professionals

As of now questions will only covers IAmTimCory video tutorials for C#.

**Project overview**

This is a summary of critical information relevant to the project,

Primary user contacts,

Technologies: Web based application

tools used :

* Visual studio 2022 – Community edition
* SQL server 2019
* Github
* Jira

**Design decisions**

This is an overview of critical decisions related to design and architecture that the team made throughout the project. Document your design decisions throughout other artifacts, such as system overviews and source code.

* **Structure: Windows Forms applications and class library**
* **Database: SQL and/or Text file**
* **Users: One at a time on one application**
* **Key concept** 
  + **Email**
  + **SQL**
  + **Custom Events**
  + **Error handling**
  + **Logs**
  + **Interfaces**
  + **Random order**
  + **texting**

**Database design**

<TODO> insert database design diagram here.

Database tables

TOPICS

|  |  |  |  |
| --- | --- | --- | --- |
| TopicID | TopicName | VideoName | VideoLink |

QUESTIONS

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| QuestionID | Question | Answer | TopicID | Explanation | OptionA | OptionB | OptionC | OptionD |

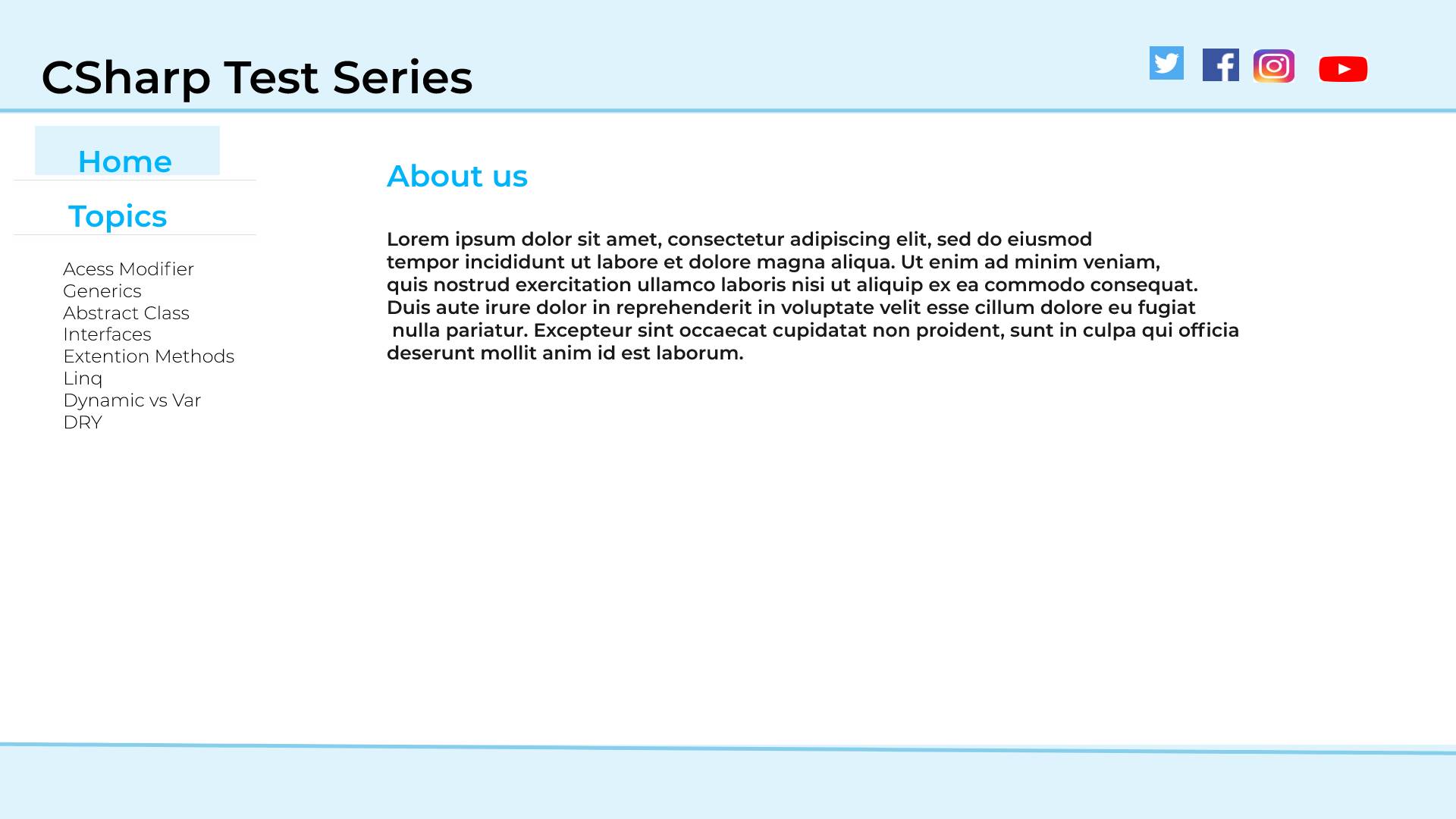


Database Path - C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\

**Website layout:**

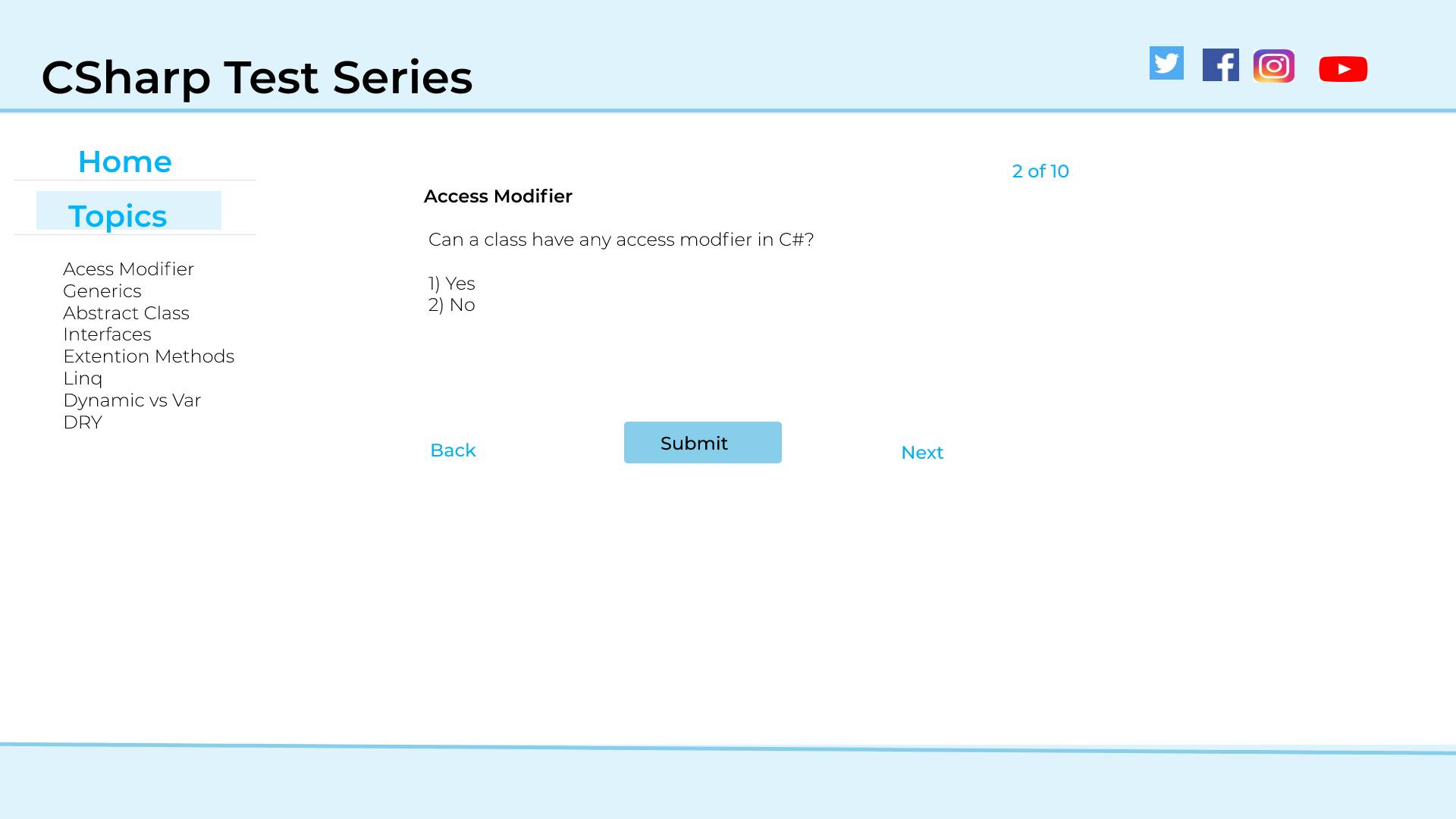
Home Page: This is the first page that user lands on.

* There could be a log on the left side of banner
* Upper banner will have
  + Logo
  + Name of the applcition
  + Link to twitter, Facebook, Instagram and youtube of IAmTimCorey
* Left menu will have a Home and Topics link. Topics link on expanding will show all the topics that users can take test on
* Middle section will have About us details



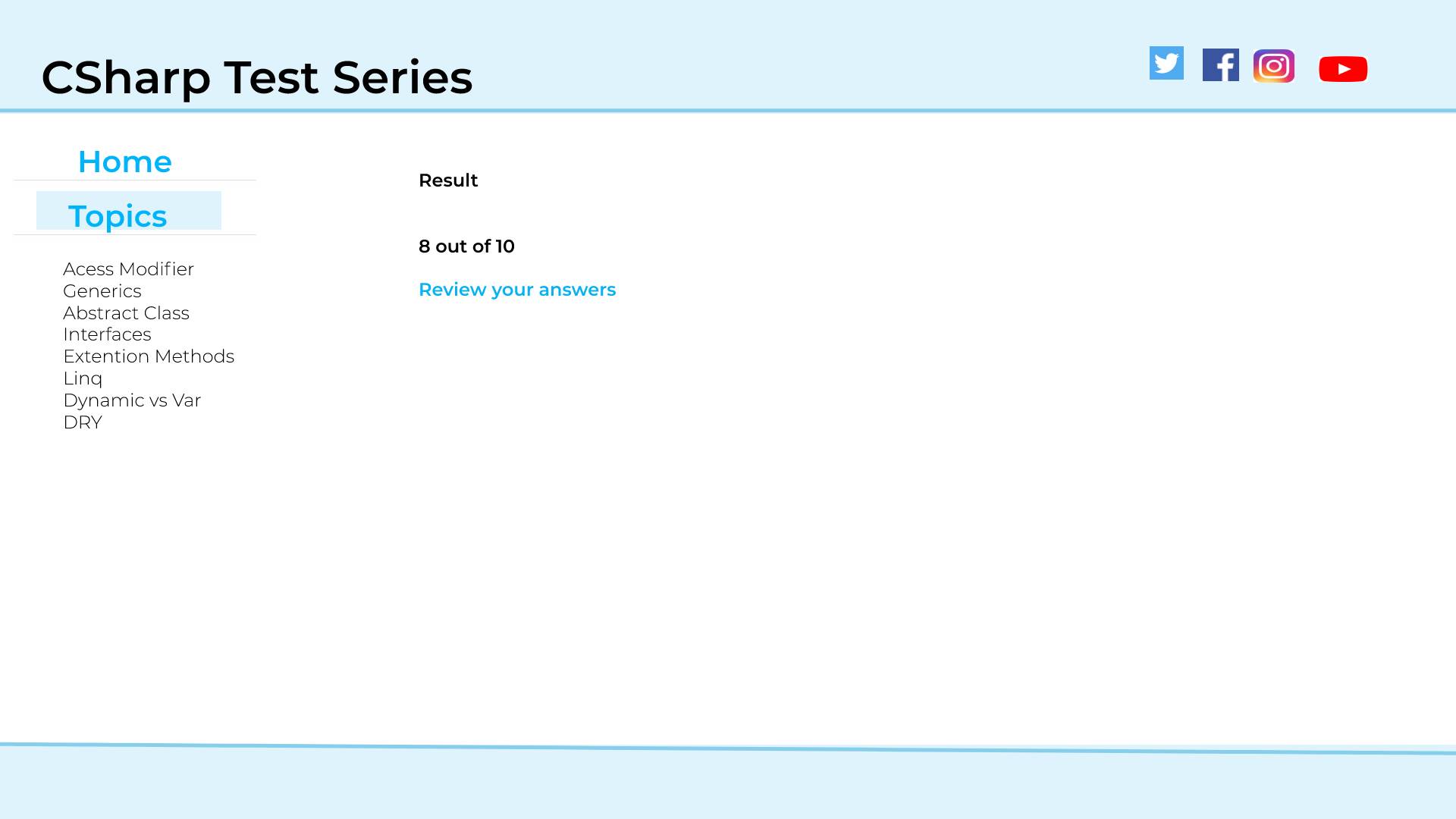
Test Page: Users can click on any of the topic on the left menu and take the test. Below show a question, options, Back link, Next link and a submit button.

* Options will be radio buttons
* When user clicks on Back link he will go to previous question
* When user clicks on Next link he will go to previous question
* When user clicks on Submit button his answer will be recorder and he will be directed to next question
* Users answer for the current question will be lost if he click on Next/Back button. It will be recorded only on submit
* There will be a counter at the top right of the screen showing current question count out of total question count

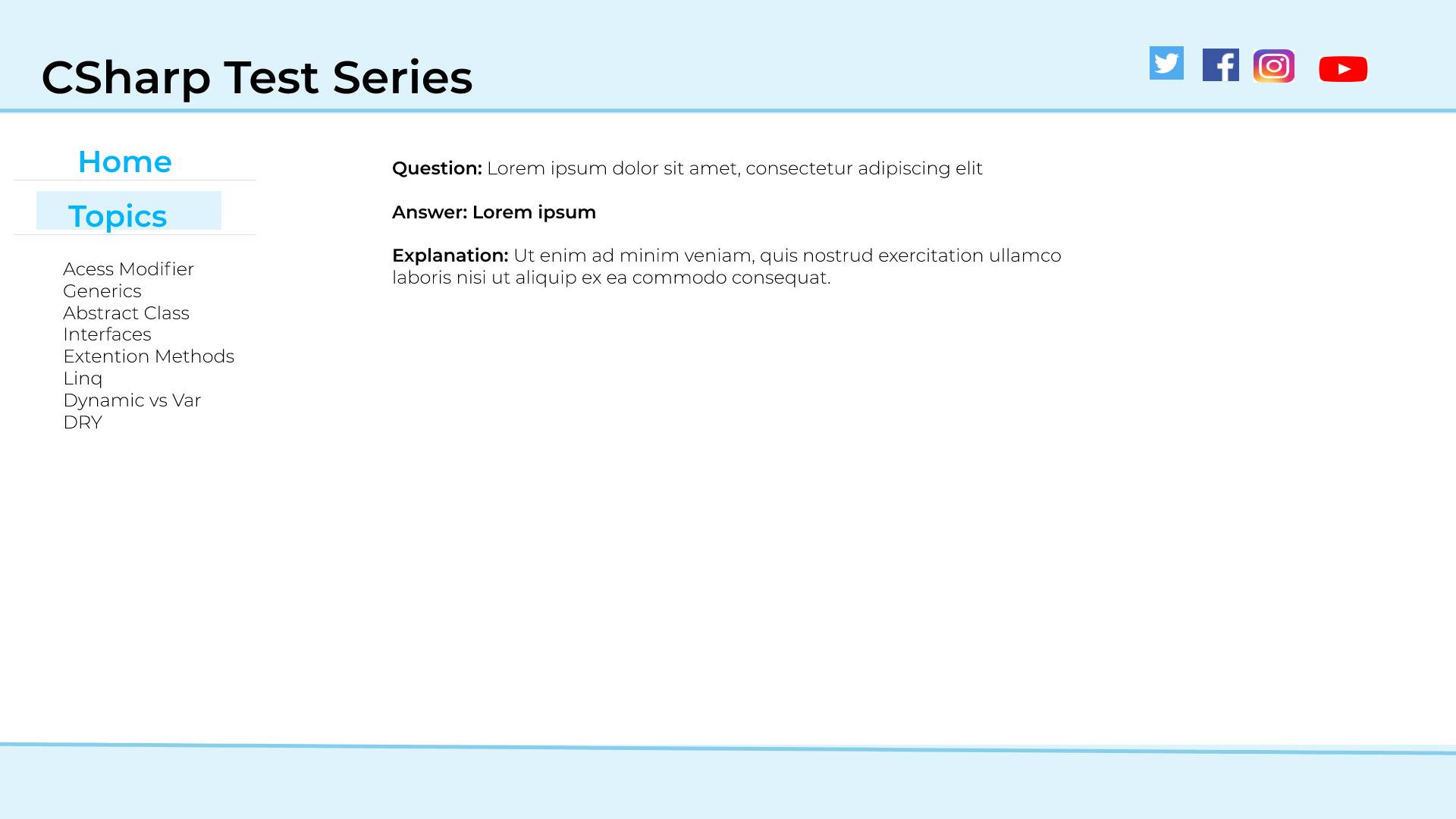


Result Page: After completing all the questions user will be redirected to results page.

* Uses will show count of correct answer out of all the questions
* User can click on the review page to review his assessment



Review Page: On review page user can see the entire question with the correct and answers and the explanation.



**Review Questions:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Questions | Answers | Additional points |
|  | What will user see on the Home Page |  |  |
|  | Is the login required to take the test |  |  |
|  | Are we going to store the individual test results |  |  |
|  | Can user contributes questions |  |  |
|  | What if the topic list grows that cannot be accommodated in a screen |  |  |
|  | Will the content on the home page be static or dynamic |  |  |
|  | What will happen I user is on the first pages and click the back button | There will not be any Back link and the Next link for first and last question respectively. |  |
|  | On the review page how to highlight wrong and right answers | TBD |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Operations documentation**

<TODO start - Delete these lines in the final version>This is usually a description of the dependencies that your system is involved with, references to backup procedures, troubleshooting guidelines, etc. Your operations department likely has a standard format for this type of documentation. <TODO End>

**Provide Github commands here**

**Provide Jira link here**

**Steps to use the application**

[**Requirements documents**](https://www.nuclino.com/articles/product-requirements-document)

It's an overview of what the system does, including use cases, user stories, or essential user interface prototypes, etc. Aim to capture the requirements as executable specs.

**Support documentation**

This includes the training materials specific to support staff, trouble-shooting guides, etc. Like the operations department, the support team may have standard templates or examples that you can work from.

**System documentation**

Provide an overview of the system, including technical architecture, business architecture, and high-level requirements. It helps ensure that if the development team leaves, critical information is left behind.

**User documentation**

t includes reference manuals and support guides for the users. Keep it simple and easy to understand. the solution design is flawed if you need to extensively train your users on how to use it.