

# **OBJECTIVE QUESTION GENERATOR APP**

Major Project Report submitted in partial fulfilment of the requirement for the degree of Bachelor of

Technology

in

Computer Science and Engineering By

Arshdeep Singh 181390

**UNDER THE GREAT SUPERVISION OF**

Dr. Himanshu Jindal



Department of Computer Science & Engineering and Information Technology

**Jaypee University of Information Technology, Wagnaghat, 173234,  
Himachal Pradesh, INDIA**

## CONTENTS

<b>Title</b>	<b>Page No.</b>
Declaration	III
Certificate	IV
Acknowledgment	V
1. Introduction	6
1.1 About the company	
1.2 About Internship	
1.3 Problem Statement	
1.4 Objective	
1.5 Methodology	
1.6 Project Overview	
2. Literature Review	9
3. System Development	10
3.1 Logical Architecture	
3.2 Physical Architecture	
3.3 Application Architecture	
3.3.1 Subsystems	
3.3.2 Use Case Diagram	
3.3.3 Modelling	
3.3.4 Layering and Partitioning	
3.3.5 Coding and UI Standards	
3.3.6 Developing Environment	
3.3.7 Software	
3.3.8 Deployment	

4. User Stories	13
5. Component Architecture	18
5.1 Class Design Specification	
5.1.1 Data Access Layer	
5.1.1.1 Class Identification	
5.1.1.2 Methods	
5.1.2 Service Layer	
5.1.2.1 Class Identification	
5.1.2.2 Methods	
5.1.3 Presentation Layer	
5.1.3.1 Components	
5.1.3.1.1 User Components	
5.1.3.1.2 Preview Components	
5.1.3.1.3 API Components	
5.2 Design of Algorithms	
5.3 Database Design	
5.4 Optimization of Algorithms and Data Access	
5.5 Error Messages	
5.6 Assumptions	
6. Operational Management	35
6.1 Performance Engineering	
6.2 Exception Handling	
6.3 Security Mechanism	
6.4 State and Session Management	
6.5 Data Access Mechanism	
7. Database Management	36
7.1 Data Model	
8. Non-Functional Requirement	36
8.1 Security	
8.2 Availability	
8.3 Scalability	
9. Images of working project.	37
10. Conclusion	49

## II Candidate's Declaration

I hereby declare that the work presented in this report entitled “ Objective Question Generator App” in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering/Information Technology submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Wagnaghat is an authentic record of my own work carried out over a period from February 2022 to May 2022 under the supervision of  
Dr. Himanshu Jindal  
Department of Computer Science and Information Technology  
Jaypee University of Information Technology

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Arshdeep Singh  
181390

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Himanshu Jindal  
Department of Computer Science and  
Information Technology  
Jaypee University of Information Technology  
Dated:

Rohini N  
Senior Educator at Education,  
Training and Assessment,  
Infosys Ltd.  
(Digital signature unavailable, as  
per Confidentiality Policy)

### III

## CERTIFICATE

This is to certify that the work which is being presented in the project report titled  
“OBJECTIVE QUESTION GENERATOR APP”  
in partial fulfilment of the requirements for the award of the degree of B.Tech in  
Computer Science And Engineering and submitted to the Department of Computer  
Science And Engineering, Jaypee University of Information Technology,  
Waknaghat is an authentic record of work carried out by **Arshdeep Singh**  
**(181390)** during the period from February 2022 to May 2022 under the supervision  
of Dr. Himanshu Jindal, Department of Computer Science and Engineering, Jaypee  
University of Information Technology, Waknaghat.

The above statement made is correct to the best of our knowledge.

**Dr. Himanshu Jindal**  
Assistant Professor(SG)  
Computer Science & Engineering and Information Technology  
Jaypee University of Information Technology, Waknaghat

## IV

### ACKNOWLEDGEMENT

Firstly, We express our heartiest thanks and gratefulness to Almighty God for His divine blessing that makes it possible to complete the project work successfully. We would also like to thank Jaypee University of Information Technology and the Department of Computer Science and Information Technology for giving us this opportunity.

We are grateful and wish our profound indebtedness to Supervisor Dr. Himanshu Jindal, Associate Professor, Department of CSE Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge & keen interest of our supervisor in the field of .Net framework and Angular helped us in carrying out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express my heartiest gratitude to Dr. Himanshu Jindal, Department of CSE, for his kind help to finish our project.

We would also generously welcome each one of those individuals who have helped us straightforwardly or in a roundabout way in making this project a win. In this unique situation, We might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, We must acknowledge with due respect the constant support and patience of our parents, guardians and siblings.

Arshdeep Singh

# **1. Introduction**

## **1.1 About the Company**

After the end of the 7th semester, various companies visited the college to enroll students. One such company was Infosys Ltd., where we selected the profile of the System Engineer. We were offered an internship program by Infosys before the full-time role and completing the internship is required for us to have a full-time role with the company. The internship will be approximately 13 weeks containing various sessions, webinar, Lex online course (Infosys online course), Lex platform test and project. Infosys is one of the top IT companies in India. Infosys employees are about 3 Lac people and employ about 20k young people every year from India. Infosys also hires in various countries around the world.

Infosys offers a variety of roles in the company as an engineer, designer, inspector and manager in the company, but, before a friend everyone has to complete an apprenticeship and after a period of apprenticeship there is a one-year probationary period for colleagues. to join a company.

The allocation of domain is a system developer training for Infosys trainees, but sometimes it depends on practice testing as well.

## **1.2 About The Internship**

Our internship at Infosys was conducted in online mode due to covid situation. The internship started on 7th February 2022. For the first two weeks i.e., from 7th -18th February 2022 we had webinars and orientation with our HRs about the company, Mysore Development Center (DC), all the location of other Development Center, salary accounts and interaction sessions. From 21st – 25th February we were given introduction to topics and in which order we were going to learn them. Also, we learned about that after covering few topics we were going to have their hand written exams for evaluation. The topics were Basic C sharp Programming, Microsoft SQL, Entity Framework Core, Developing Service Layer using ASP.NET Web API, HTML, CSS and Angular for .NET Developers. Now for the next two weeks i.e., from 28th February to 11th March we were taught about Basic C Sharp Programming [1], about its fundamentals, data types and basic object-oriented programming. Also, the assignments were given so that we can get

familiar with the language. Then from 14th March to 19th March we were taught about Microsoft SQL[8]. In this topic we were basically taught about stored procedures and functions like how to create them and how to use them. Then from 21st March to 25th March we studied about Entity Framework Core (EF Core) [2]. The basic functionality of EF Core is to use CRUD operations on the stored procedures and functions created in the database, connect them to API. Then from 28th March to 1st April we studied about Developing Service Layer using ASP.NET Web API [3] which helped us to create web services or Restful API so that we can create client services. Then from 1st April to 7th April we studied about HTML, CSS, Bootstrap and Angular for .NET Developers [4] [5] [6] [7] which helped us to create the presentation layer or the front end of application/websites. The details about these topics are discussed in literature survey. Then from 8th April to 10th May we got a team of 6 people and were given a capstone project by the name 'Question Generator App' where we needed to create a working project from the scratch with the all the topics we learned till now and we were evaluated after every sprint. The sprint duration was 10 days. We needed to decide the user stories for this project and then positively start work upon them.

### **1.3 Problem Statement**

In these covid times the exams are shifting more and more towards online medium and objective examination is the new trend so one needs an app to search questions on any certain topics and create a question paper out of it. Question generator app can be used by a teacher to test his students or a student can generate set of questions for himself to test his/her skills with easy.

### **1.4 Objective**

The teams objective was to create an app using which one can search an objective question on any topic of his/her choice and add them to a previewable file where the question can be deleted or edited and later the set of questions could get downloaded as well.



## **1.5 Methodology**

For the Question Generator system, the front-end technology is Angular. The data access tier is built with EntityFrameworkCore and services are built with ASP.NET Web API Core backed by C# in order to leverage the benefits of object-oriented programming such as platform independence and robustness. The UML diagrams are used for the class design.

At the back-end Microsoft SQL Server 2019 database is used for persistent storage of data. The tables are designed in such a way that high performance is ensured when any operation is performed in the database.

## **1.6 Project Overview**

This application will generate possible question along with multiple options for given set of code snippet. The user can save or discard the questions as per their requirement. The questions must be saved in library if the user saves the question. Also, the user should be able to edit the generated questions. The user should be able to download the questions as PDF, Excel whenever needed.

## 2. Literature Review

In this project we have made use of various tech stacks such as C#, ENTITY FRAMEWORK CORE, ASP.NET, HTML, CSS, Bootstrap, Angular and Azure Devops [9]. These technologies have been around from quite some time now. Microsoft first-time used C# in 1988 [10] similarly HTML was invented in 1991. Microsoft's .NET framework is a proprietary software owned by microsoft which is predominantly used for web and game application developments. The frontend of the project was developed using angular which is a typ-script based free and open source web application which was developed by google and a community of individuals and corporations in 2010 [11].

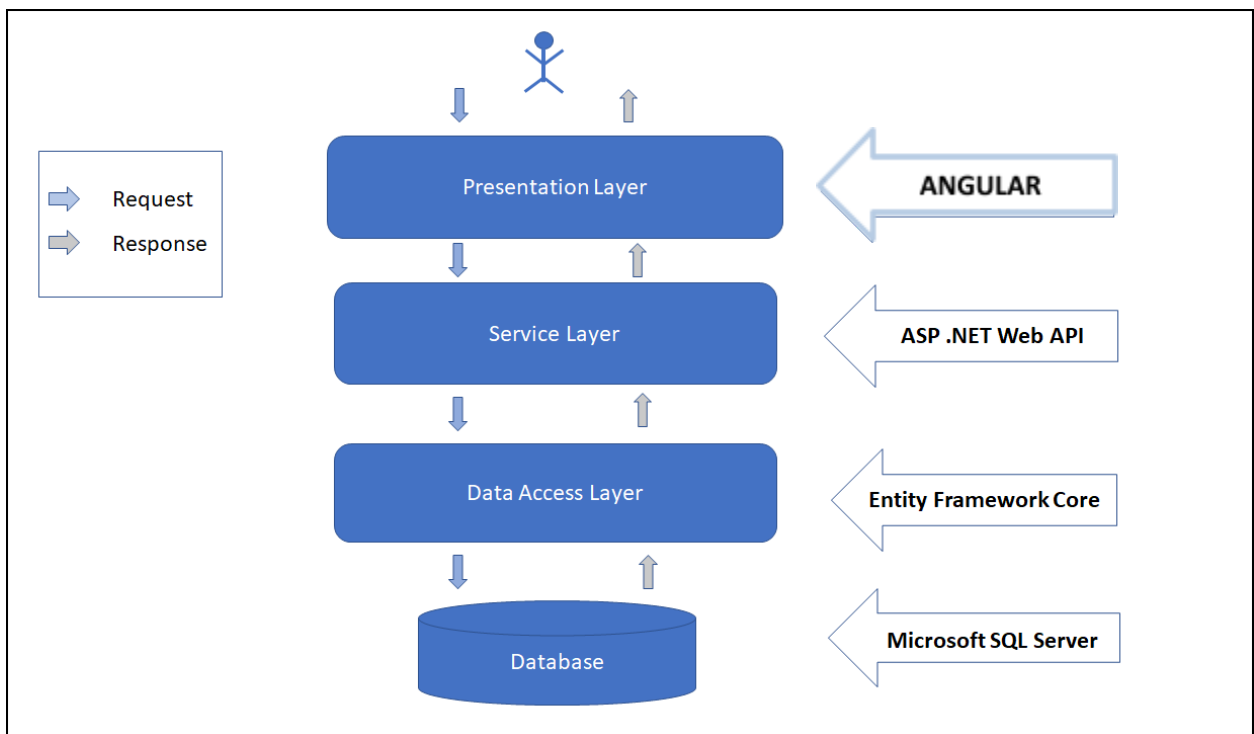
People have created various previous projects to generate objective questions but most of them provide paid services such as rapid refresh by ED APP [12] , some have very poor UI such as EasyTestMaker [13] and ClassMarker [15], some posses very large range of services which becomes very confusing for user such as Typeform [14] , and various other application available on different tech stacks like Proprofs [16] and FyreBox [17].

Many of the projects present on the internet have made use of machine learning and deep learning techniques such as Naive Bayes, Sentimen analysis for generating questions from a certain keyword. There have been various project developed which have been contributed to open source such as Github like one by Suraj Patil in which he has used Natural language processing [18]. There is also one project by Kristian Vachev where he has used Naive Bayes and Word embeddings to generate multiple choice questions from text [19]. You can aslo see few other such projects such as one by Yuxi Xie where he has used semantic graph and content selection to generate objective questions [20].

### 3. System Architecture

#### 3.1 Logical Architecture

The following diagram shows the big picture of the Question Generator Architecture.



## 3.2 Physical Architecture

Type	Configuration	Installed Software
Web server	IIS Server	IIS 12.0
Database server	MS SQL Server 2019	MS SQL Server 2019
Front end development environment	Angular	Angular CLI, Node Js

## 3.3 Application Architecture

### 3.3.1 Subsystems

- Register
- Reset Password
- Login
- View Questions
- Search Questions
- Save Selected Questions
- Modify Questions
- Preview
- Delete from Preview
- Download Questions
- Feedback
- Logout

### 3.3.2 Modelling

Based on the use case model defined in the requirement engineering face. The following models need to be developed:

- Object Model.
- Deployment Model.

### **3.3.3 Layering and Partitioning**

This application follows three-tiered architectural approach.

The architectural followed has 3 layers:

- Front end or Presentation Layer
- Service Layer
- Data Access Layer

Following are the functionalities implemented in each of these layers:

Presentation Layer: Responsible for displaying the data and interacting with the end user.

Service Layer: Responsible for designing and implementation of the user-defined APIs.

Data access Layer: Responsible for reading/writing data from/to database.

### **3.3.4 Coding and UI Standards**

Coding standard of Question Generator development is also based on the dotnet coding standards.

### **3.3.5 Development Environment**

For the development of the Question Generator system of the following specification are used:

- RAM: 8 GB
- HDD: 100 GB
- Network: Above 1 MBps
- Operating system: Windows 10

### **3.3.6 Software**

- Visual Studio 2019 IDE
- .NET Core
- Microsoft SQL Server 2019

### **3.3.7 Deployment**

For the deployment of the Question Generator system of the following specification are used:

RAM: 8 GB

HDD: 100 GB

Network: Above 1 MBps

Operating system: Windows 10

#### 4. User Stories

<b>USER STORY ID</b>	#1	<b>PRIORITY</b>	Must Have
<b>User Story Description</b>			
As a user, I want to register the user first, so that I can use the services provided by the app.			
<b>Acceptance Criteria</b>			
1.Password: 8-15 characters 2.Email must be valid 3.Username must be unique. 4. First and last name are required.			

<b>USER STORY ID</b>	#2	<b>PRIORITY</b>	Must Have
<b>User Story Description</b>			
As a user, I want to login at all times, so that I can access my saved work.			

**Acceptance Criteria**

For login user must be registered.

<b>USER STORY ID</b>	#3	<b>PRIORITY</b>	Must Have
----------------------	----	-----------------	-----------

**User Story Description**

As a user, I want to enter a code snippet or sample question, so that I can generate a similar question based on text.

**Acceptance Criteria**

1. User must be logged in.
2. If no relative keyword is found data not found must be returned.

<b>USER STORY ID</b>	#4	<b>PRIORITY</b>	Must Have
----------------------	----	-----------------	-----------

**User Story Description**

As a user, I want to save the question, so that I can preview the question when needed.

**Acceptance Criteria**

1. At least one question must be displayed for save to happen.

<b>USER STORY ID</b>	#5	<b>PRIORITY</b>	Must Have
----------------------	----	-----------------	-----------

**User Story Description**

As a user, I want to preview all the questions I have saved, so that I can make the necessary changes if required.

**Acceptance Criteria**

1. At least one question must be saved for preview to happen.

<b>USER STORY ID</b>	#6	<b>PRIORITY</b>	Must Have
----------------------	----	-----------------	-----------

**User Story Description**

As a user, I want to rearrange or edit all the questions, so that I can have questions as per my choice.

**Acceptance Criteria**

1. To rearrange the question order at least 2 questions should be selected and for editing the question even one is enough.

<b>USER STORY ID</b>	#7	<b>PRIORITY</b>	Must Have
----------------------	----	-----------------	-----------

**User Story Description**

As a user, I want to discard any questions from the previewed question paper, so that I can have my desirable set of questions.

**Acceptance Criteria**

1. At least one question must be in preview to be discarded.



--

<b>USER STORY ID</b>	#8	<b>PRIORITY</b>	Must Have
<b>User Story Description</b>			
As a user, I want to download the previewed set of questions, so that I can access it on my local machine.			
<b>Acceptance Criteria</b>			
1. There must be one question so that it can be downloaded.			

<b>USER STORY ID</b>	#9	<b>PRIORITY</b>	Must Have
<b>User Story Description</b>			
As a user, I want to save the data, so that I can avoid data loss.			
<b>Acceptance Criteria</b>			
1. The data must be saved every 20 minutes.			

<b>USER STORY ID</b>	#10	<b>PRIORITY</b>	Must Have
<b>User Story Description</b>			
As a user, I want to logout from my profile, so that no one can access my profile without my consent.			
<b>Acceptance Criteria</b>			
<ol style="list-style-type: none"> <li>1.The User must be logged in to logout.</li> <li>2.The user is inactive for more than 1 hour and the user will be automatically logged out.</li> </ol>			

<b>USER STORY ID</b>	#11	<b>PRIORITY</b>	Must Have
<b>User Story Description</b>			
As a user, I want to reset my password, so that I am able to login with my new password.			
<b>Acceptance Criteria</b>			
<ol style="list-style-type: none"> <li>1.User must be registered</li> </ol>			

<b>USER STORY ID</b>	#12	<b>PRIORITY</b>	Must Have
<b>User Story Description</b>			
As a user, I should be able to give feedback, so that changes could be made and issues could be brought in notice.			

## Acceptance Criteria

1. User must be logged in.

## 5. Component Architecture

The component architecture of the system has been captured using object model and deployment model. The details of these models are captured in the subsequent sub sections.

### 5.1 Class Design Specification

#### 5.1.1 Data Access Layer

##### 5.1.1.1 Class Identification

<b>Class Name</b>	QuestionDBRepo
<b>Class Description</b>	It contains the DAL logic and acts as an intermediate between the database and the service layer.
<b>Class Inheritance</b>	NA
<b>Classes Referenced</b>	Models.Feedback, Models.History, Models.Paper, Models.PapaerId, Models.Quest, Models.Question, Models.ResetPassword, Models.User, Models.questionDB1Context.
<b>Sub-system</b>	NA
<b>Class Type</b>	Complete

##### 5.1.1.2 Methods

<b>Method</b>	public bool AddUser(string firstname, string lastname, string email, string password)
<b>Purpose</b>	To register user details using stored procedure from database.
<b>Returns</b>	Status indicating if the logic is executed successfully or not.

<b>Method</b>	public bool LoginUser(string email, string password)
<b>Purpose</b>	To Login User using scalar-valued function from database.
<b>Returns</b>	Status indicating if the logic is executed successfully or not.

<b>Method</b>	public bool ResetPassword(string email, string oldpassword, string newpassword)
<b>Purpose</b>	To Update the user password in the database using stored procedure.
<b>Returns</b>	Status indicating if the logic is executed successfully or not.

<b>Method</b>	public List<Question> GetQuestion()
<b>Purpose</b>	To view all questions from database.
<b>Returns</b>	questionList of all stored questions.

<b>Method</b>	public List<Question> FetchQuestions(string search)
<b>Purpose</b>	To search questions according to the categories stored in database.
<b>Returns</b>	Question with question if the logic is executed successfully and null if not executed .

<b>Method</b>	public int SaveQuestion(string emailid, string questionid)
<b>Purpose</b>	To store user selected questions.
<b>Returns</b>	result indicating if the logic is executed successfully or not.

<b>Method</b>	public int DeleteQuestions(string emailid, string questionid)
<b>Purpose</b>	To delete selected questions from preview.
<b>Returns</b>	Result indicating if the logic is executed successfully or not.

<b>Method</b>	public List<Question> PreviewQuestion(string emailid)
<b>Purpose</b>	To show user selected questions in preview.
<b>Returns</b>	previewquestions indicating if the logic is executed successfully and null if not executed.

<b>Method</b>	public bool AddNewQuestion(string category, string question, string option1, string option2, string option3, string option4, string answer)
<b>Purpose</b>	To give user functionality of adding questions according to their choice.
<b>Returns</b>	Status indicating if the logic is executed successfully or not.

## 5.1.2 Service Layer

### 5.1.2.1 Class Identification

<b>Class Name</b>	UsersController
<b>Class Description</b>	It contains the APIs and acts as an intermediate between the Data Access Layer and the Presentation Layer.
<b>Class Inheritance</b>	NA
<b>Classes Referenced</b>	QuestionDBRepo, Models.class, Models.Delete, Models.Feedback, Models.History, Models.Paper, Models.Paperid, Models.Quest, Models.Question, Models.ResetPassword, Models.User, Models.ValidateUser.
<b>Sub-system</b>	NA
<b>Class Type</b>	Complete

### 5.1.2.2 Methods

<b>Method</b>	public bool AddUsers (QuestionWebApi.Models.User user)
<b>Purpose</b>	To register the new user by invoking the DAL method.
<b>Returns</b>	Message indicating (true if yes and false if error is there).

<b>Method</b>	public bool Login (QuestionWebApi.Models.User user)
<b>Purpose</b>	To validate the credentials prompted by the user during login by invoking the DAL method.
<b>Returns</b>	Status indicating (true if yes and false if error is there).

<b>Method</b>	public bool ResetPass(QuestionWebApi.Models.ResetPassword obj)
<b>Purpose</b>	To validate the credentials prompted by the user and updating the new password by invoking the DAL method.
<b>Returns</b>	Status indicating (true if yes and false if error is there).

<b>Method</b>	public bool SaveQuestions(QuestionWebApi.Models.Class c)
<b>Purpose</b>	To save user selected questions by invoking the DAL method.
<b>Returns</b>	Message indicating (true if yes and false if error is there).

<b>Method</b>	public bool DeleteQuestion(QuestionWebApi.Models.Delete delete)
<b>Purpose</b>	To deleted user saved questions by invoking the DAL method.
<b>Returns</b>	Message indicating (true if yes and false if error is there).

<b>Method</b>	public JsonResult GetQuestion()
<b>Purpose</b>	To view all questions by invoking the DAL method.
<b>Returns</b>	Return questions in JSON format.

<b>Method</b>	public JsonResult FetchQuestions(string search)
<b>Purpose</b>	To search specific questions by invoking the DAL method.
<b>Returns</b>	Return fetchQuestions in JSON format.

<b>Method</b>	public bool SaveFeedback(QuestionWebApi.Models.Feedback feedback)
<b>Purpose</b>	To save user feedback by invoking the DAL method.
<b>Returns</b>	Message indicating (if successful true else false if any error occurs).

<b>Method</b>	public JsonResult PreviewQuestions(string emailid)
<b>Purpose</b>	To show previews of user selected questions by invoking the DAL method.
<b>Returns</b>	Fetchquestion in JSON format.

### 5.1.3 Presentation Layer

#### 5.1.3.1 Components

##### 5.1.3.1.1 User Components

- User Registration.

<b>Method</b>	addUser(firstName: string, lastName: string, emailId: string, password: string):Observable<boolean>
<b>Purpose</b>	To register the user details provided by the user using the User Service method.

<b>Returns</b>	Success redirects to login and error gives alert.
----------------	---

- User Login.

<b>Method</b>	login(emailId:string,password:string):Observable<string>
<b>Purpose</b>	To login the user by the credentials provided by the user using the User Service method.
<b>Returns</b>	Success redirects to View questions in app and error gives alert.

- Save Question.

<b>Method</b>	SaveQuestion(emailId: string,quesId: string ): Observable<boolean>
<b>Purpose</b>	To save the selected questions of user using the User Service method.
<b>Returns</b>	Success gives an success alert and error gives error alert.

- Reset Password.

<b>Method</b>	resetPassword(email: string, oldPassword: string, newPassword: string ): Observable<boolean>
<b>Purpose</b>	To change password of the user using the User Service method.
<b>Returns</b>	Success redirects to login and error gives error alert.

- Delete Selected Questions from Preview.

<b>Method</b>	deleteCartProduct(quesId: string, emailId: string): Observable<boolean>
<b>Purpose</b>	To delete the user selected questions from preview by user using the User Service method.
<b>Returns</b>	Success gives alert and error gives error alert.

- Feedback

<b>Method</b>	Feedback (feedback: string): Observable<boolean>
<b>Purpose</b>	To store the feedback provided by the user using the User Service method.
<b>Returns</b>	Success redirects to ViewQuestions and error gives error alert.



- Error Handler

<b>Method</b>	errorHandler(error: HttpResponse)
<b>Purpose</b>	To handle any error, occur to the user while using application using the User Service method.
<b>Returns</b>	Error gives error alert.

### 5.3.1.2 Preview Components

- Get Questions

<b>Method</b>	getQuestions(): Observable<any>
<b>Purpose</b>	To get all questions from the database using the Preview Service method.
<b>Returns</b>	Return tempVar.

- Preview Questions

<b>Method</b>	previewQuestions(emailId: string): Observable<any>
<b>Purpose</b>	To get all questions from the database using the Preview Service method.
<b>Returns</b>	Return tempVar.

- Set Questions

<b>Method</b>	previewQuestions(emailId: string): Observable<any>
<b>Purpose</b>	To get all questions from the database using the Preview Service method.
<b>Returns</b>	Return tempVar.

- Save Preview

<b>Method</b>	view(question: any)
<b>Purpose</b>	To get all user saved questions from the database using the Preview Service method.
<b>Returns</b>	Return preview list of questions.

- Remove Preview Item

<b>Method</b>	removePreviewItem(question: any)
<b>Purpose</b>	To delete user saved questions from the preview using the Preview Service method.
<b>Returns</b>	Return preview list of questions.

- Error Handler

<b>Method</b>	errorHandler(error: HttpResponse)
<b>Purpose</b>	To handle any error, occur to the user while using application using the User Service method.
<b>Returns</b>	Error gives error alert.

### 5.3.1.3 API Components

- Get Questions

<b>Method</b>	getQuestions(): Observable<IQuestion[]>
<b>Purpose</b>	To view all questions from the database using the API Service method.
<b>Returns</b>	Return tempVar.

- Fetch Questions

<b>Method</b>	getQuestions(): Observable<IQuestion[]>
<b>Purpose</b>	To view specific searched questions from the database using the API Service method.
<b>Returns</b>	Return IQuestion array.

- Error Handler

<b>Method</b>	errorHandler(error: HttpResponse)
<b>Purpose</b>	To handle any error, occur to the user while using application using the User Service method.
<b>Returns</b>	Error gives error alert.

## 5.2 Design of Algorithms

This system is using standard algorithms for functionalities including searching, sorting, retrieving the data, deleting the data, adding the data. There is no special critical algorithm used in Question Generator.

## 5.3 Database Design

### 5.3.1 Overview of tables

- Db.users

Table used to store user's data for profile and login.

- Db.questions

Table to store questions along with their category, options and answers.

- Db.paper

Table to store userId with their unique paperId.

- Db.Quest

Table to store user selected questions for their question paper.

- Db.Feedback

Table to store users feedback.

## **5.3.2 Overview of stored procedures and functions**

### **5.3.2.1 Stored Procedures**

a. Dbo.uspAddUser

Procedure that adds user registration data to dbo.users table.

b. Dbo.resetPass

Procedure that updates the user passwords in dbo.user table.

c. Dbo.AddNewQuestion

Procedure to store user added questions to dbo.question table.

d. Dbo.DeleteQuestion

Procedure to delete user selected questions for question paper from dbo.Quest table.

e. Dbo.SaveQuestion

Procedure to save user selected questions to dbo.Quest table.

### **5.3.2.2 Functions**

a. Dbo.ChkEmailId

Function to validate if email is correctly specified or not and storing it in table dbo.users.

b. Dbo.ufn\_ValidateUserCredentials

Function to validate user credentials.

c. Dbo.ID

Function to store QuesPaperId for every new row inserted in dbo.paper table.

d. Dbo.quesID1

Function to store QuesId for every new row inserted in dbo.questions table.

e. Dbo.userID2

Function to store UserId for every new row inserted in dbo.users table.

### 5.3.3 Database Objects

#### 5.3.3.1 Tables

##### i) Users

<b>Description</b>	This table contains the details of the users of the system
--------------------	--

#.	Primary Key	Column Name	Data Type	Size	Nulls	Unique	Default	Check Constraint
1.	Y	UserID	CHAR	4	N		Y	
2.		FirstName	VARCHAR	20	N			
3.		LastName	VARCHAR	20	N			
4.		EmailId	VARCHAR	30	N	Y		chk_email
5.		Passwd	VARCHAR	5	N			

## ii) Questions

<b>Description</b>	This table contains questions their options categories and answers of the system
--------------------	--

#.	Primary Key	Column Name	Data Type	Size	Nulls	Unique	Default	Check Constraint
1.	Y	QuesID	CHAR	4	N		Y	
2.		Category	VARCHAR	30	N			
3.		Ques	VARCHAR	4000	N	Y		
4.		Option1	VARCHAR	500	N			
5.		Option2	VARCHAR	500	N			
6.		Option3	VARCHAR	500	N			
7.		Option4	VARCHAR	500	N			

8.		Answer	VARCH AR	500	N			
----	--	--------	-------------	-----	---	--	--	--

### iii) Paper

<b>Description</b>	This table unique paperid and userid of the user
--------------------	--

#.	Primar y Key	Column Name	Data Type	Size	Null s	Uniqu e	Defaul t	Check Constraint
1.	Y	QuesPaperID	CHAR	4	N		Y	
2.		UserId	CHAR	4	N			Foreign key Users(userI d)

### iv) Quest

<b>Description</b>	This table unique paperid and question id of the user selected questions.
--------------------	---

#.	Primar y Key	Column Name	Data Type	Size	Null s	Uniqu e	Defaul t	Check Constraint
1.		QuesID	CHAR	4	N			Foreign key Questions(QuesI d)
2.		PaperId	CHAR	4	N			Foreign key paper(QuesPaper Id)

## v) Feedback

<b>Description</b>	This table unique paperid and question id of the user selected questions.
<b>Identity Column</b>	Serial No

#.	Primary Key	Column Name	Data Type	Size	Nulls	Unique	Default	Check Constraint
1.	Y	Serial_No	INT		N			
2.		Feedback	Varchar	1500	N			

### 5.3.3.2 Stored Procedures

#### i) Dbo.uspAddUser

Referenced tables	Users
Input Parameters	EmailId, Password, FirstName, LastName
Output Parameters	NA
Return Values	1 – Success 0 - failure  0- exception
Description	This procedure is used to add the details of user in the system.
Procedures/Functions Invoked	Dbo.ChkEmailId

#### ii) Dbo.resetPass

Referenced tables	Users
Input Parameters	EmailId, Previouspass, Password
Output Parameters	NA
Return Values	1– Success 0 - failure  0-exception



Description	This procedure is used to update the password in the system.
Procedures/Functions Invoked	NA

### iii) Dbo.SaveQuestions

Referenced tables	Users, Paper, Quest
Input Parameters	PaperId, result, userId
Output Parameters	NA
Return Values	1- Success 0- failure
Description	Save user selected questions.
Procedures/Functions Invoked	NA

### iv) Dbo.DeleteQuestion

Referenced tables	Users, Paper, Quest
Input Parameters	PaperId, result, userId
Output Parameters	NA
Return Values	1- Success 0- exception
Description	delete user selected questions.
Procedures/Functions Invoked	NA

### v) Dbo.AddNewQuestion

Referenced tables	Questions
Input Parameters	Category, Ques, option1, option2, option3, option4, answer
Output Parameters	NA
Return Values	1 - Success -1 - category is null

	-2 - - Ques is null -3 - if question already exists -99 – Exception
Description	delete user selected questions.
Procedures/Functions Invoked	NA

### 5.3.3.3 Functions

#### i) Dbo.ChkEmailId

Referenced Tables	Users
Input Parameters	Email
Return Values	email
Description	This function validates the email and store it in user table.

#### ii) Dbo.ID

Referenced Tables	paper
Input Parameters	Id
Return Values	Id
Description	This function generates and stores the quespaperid in paper table.

#### iii) Dbo.userID2

Referenced Tables	Users
Input Parameters	Id2
Return Values	Id2
Description	This function generates and stores the userid in user table.

#### iv) Dbo.quesID1

Referenced Tables	Questions
Input Parameters	Id1
Return Values	Id1
Description	This function generates and stores the QuesId in questions table.

## **5.4 Optimization of algorithms and data access**

The following techniques are used to optimize the database implementation:

Proper usage of index

All the tables are normalized not more than 3 NF.

## **5.5 Error Messages**

Appropriate and consistent error messages should be given.

Declarative as well as structured error handling must be used.

Try catch blocks must be used wherever required.

Error message is displayed on the screen must be consistent with respect to font position etc., across all the forms of the website.

## **5.6 Assumptions**

In order to use the system, the user must register and sign in.

User must select the questions to download and preview.

For deletion from preview there must be 1 selected question.

User can also add questions if he/she wants specific questions of their wants.

## **6. Operational Management**

### **6.1 Performance Engineering**

The system is expected to service the request of 200 users concurrently. Angular has been used to improve the user experience and performance.

### **6.2 Exception Handling**

The exception handling of Question Generator is made robust. Exceptions are identified based on the captured requirements. Exception handling is used in every method of used classes.

### **6.3 Security Mechanism**

Users must be logged in and authorized before they can access the main features of application.

### **6.4 State and Session Management**

Depending on the user email the preview of the selected questions is done.

### **6.5 Data Access Mechanism**

The data is accessed from the database according to the requirement. If any record is required from the database, then the connection to the database is created at the time of

the request and the required details are fetched. The connection string is stored in the configuration file for ease of maintenance.

## **7. Database Management**

### **7.1 Data Model**

Microsoft SQL Server uses the relational database management system all the connections to the database are made by the data access classes.

## **8. Non-Functional Requirement**

### **8.1 Security**

As the proposed system has different uses and each user performs the role specific functionalities, it should have high security and the same is implemented using Role Based Access Control (RBAC).

### **8.2 Availability**

The system, being an online one, will be available around the clock.

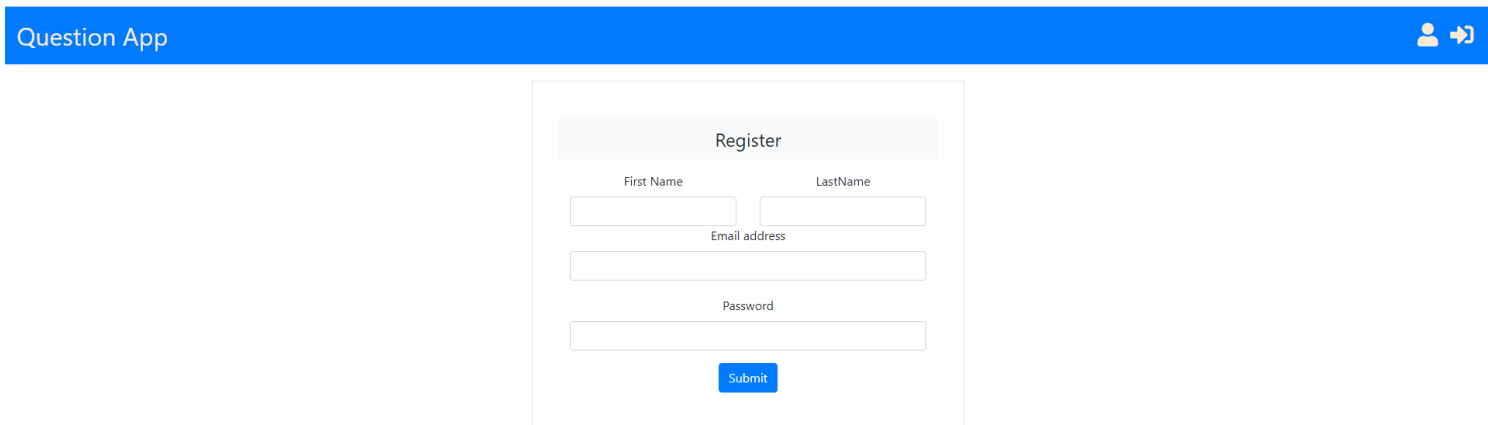
### **8.3 Scalability**

This application can handle the request from 200 users at a time without any performance degradation.

## 9. Images Of Working Project:

### A)Registration:

#### i)Layout



The image shows a mobile application interface for registration. At the top, there is a blue header bar with the text "Question App" on the left and a user profile icon with a right-pointing arrow on the right. Below the header is a registration form titled "Register". The form contains the following fields and elements:

- Two input fields for "First Name" and "LastName".
- A single input field for "Email address".
- A single input field for "Password".
- A blue "Submit" button located below the password field.

#### ii) Registration with conditions(If Fields are left empty):

### Register

First Name      LastName

Email address

Password

iii) If all conditions are satisfied:

### Register

First Name      LastName

Shivang      Mishra

Email address

shivangmishra@gmail.com

Password

\*\*\*\*\*

## B)Login:

i)Layout:

### Login

Email Id

Password

[Change Password](#)

ii)When conditions are not satisfied:

### Login

Email Id

Email id is mandatory

Password

Password is mandatory

[Change Password](#)

iii)When conditions are satisfied:



Question App

localhost:4200 says  
Login Successful

OK

### Login

Email Id

Password

[Change Password](#)

## C)Change Password:

i)Layout:

Question App

### Change Password

Email Id

Old Password

New Password

ii)When change password is successful:

The screenshot shows a web application interface. At the top left, a blue header contains the text "Question App". To its right, a notification box displays "localhost:4200 says password reset" with an "OK" button. On the far right of the header, there are icons for a user profile and a refresh/reload function. The main content area features a white modal window titled "Change Password". Inside this modal, there are three input fields: "Email Id" (containing "shivangmishra@gmail.com"), "Old Password" (masked with "\*\*\*\*\*"), and "New Password" (masked with "\*\*\*\*\*"). A blue "Reset Password" button is positioned at the bottom center of the modal.

**D)Modify question:**

i)Layout:

The screenshot displays a web application interface for modifying a question. The top blue header includes a home icon, the text "View Questions", and icons for a shopping cart, chat, and refresh. The main content area is a white form titled "Modify your own question". The form contains several input fields: "Category" and "Question" (side-by-side), "Option1", "Option2", "Option3", "Option4", and "Answer" (each on a new line). A blue "Submit" button is located at the bottom center of the form.

ii)When modification is successful:

View Questions localhost:4200 says  
Question Added Successfully

**Modify your own question**

Category:  Question: 

Option1: 

Option2: 

Option3: 

Option4: 

Answer:

## E)View Questions:

i)Layout:

View Questions 🛒 🗨️ ➡️

**Search Question**

**How do you initialize an array in C?**

a) 

b) 

c) 

d) 

correct answer : c)

**What are the advantages of arrays?**

a) 

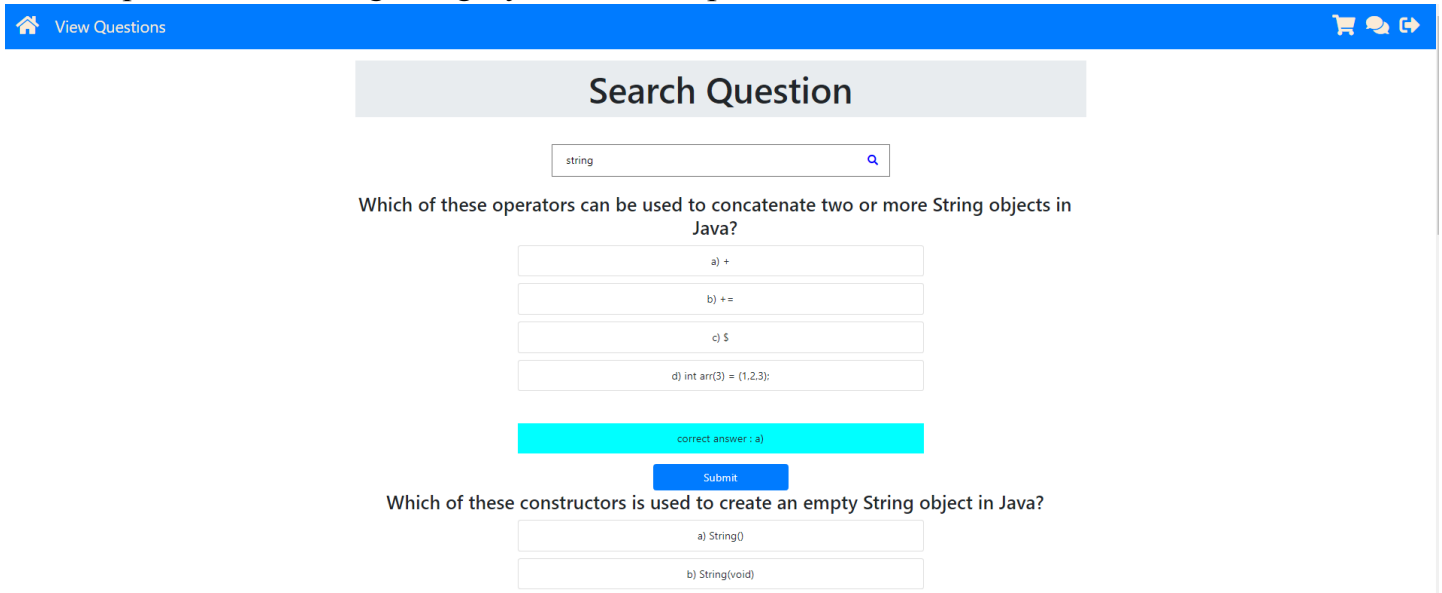
b) 

c)

## F)Search Question:

i)Searching for string category:

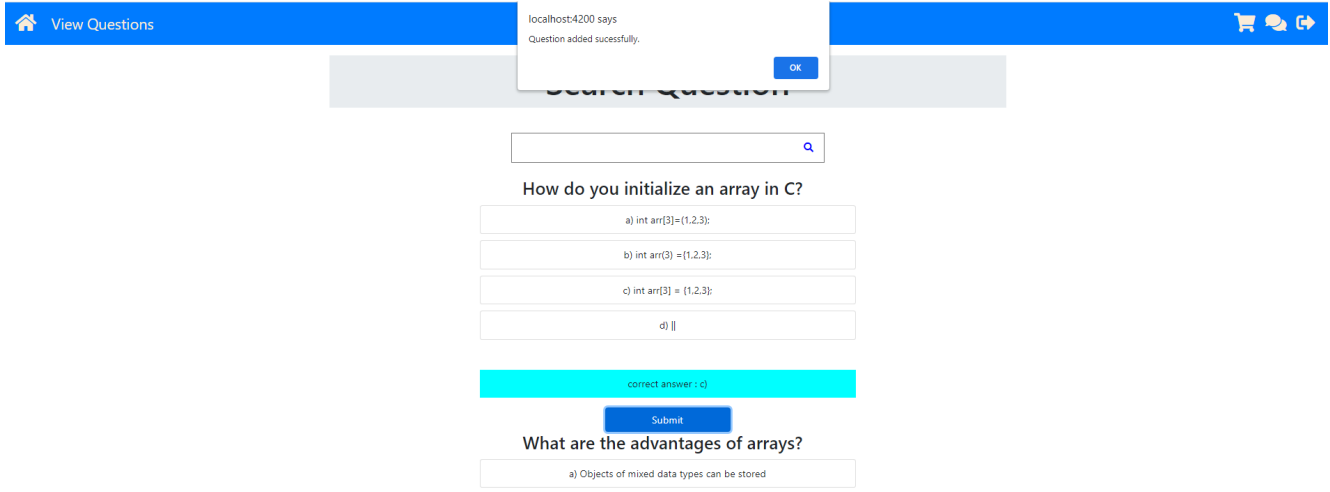
All the questions in string category will show up.



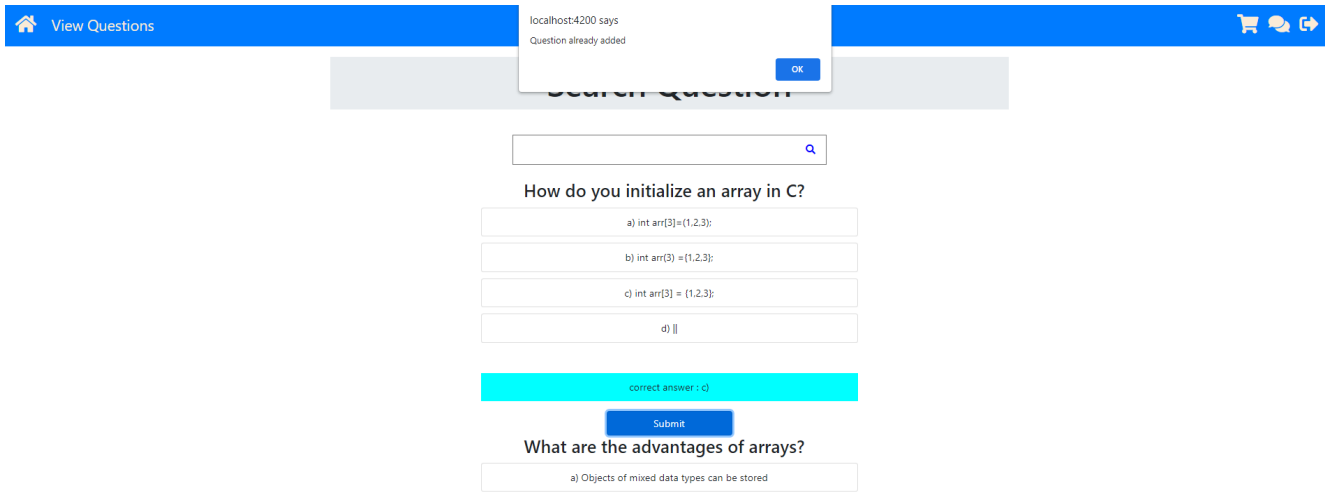
The screenshot shows a web interface for searching questions. At the top, there is a blue navigation bar with a home icon, the text "View Questions", and icons for a shopping cart, a speech bubble, and a refresh button. Below the navigation bar is a grey header with the text "Search Question". A search input field contains the word "string" and has a magnifying glass icon on the right. Below the search field, the question text reads: "Which of these operators can be used to concatenate two or more String objects in Java?". There are four radio button options: a) +, b) +=, c) \$, and d) int arr(3) = {1,2,3};. A red bar highlights the text "correct answer : a)". Below this is a blue "Submit" button. The next question text reads: "Which of these constructors is used to create an empty String object in Java?". There are two radio button options: a) String() and b) String(void).

## G)Adding question to preview:

i)When question is not added to preview it will not give any error:



ii)When question is already added to preview it will give question already added:



## H)Preview Questions:

i)Layout:

How do you initialize an array in C?

- a) `int arr[3]=(1,2,3);`
- b) `int arr(3) =(1,2,3);`
- c) `int arr[3] = {1,2,3};`
- d) `||`

Delete

Download

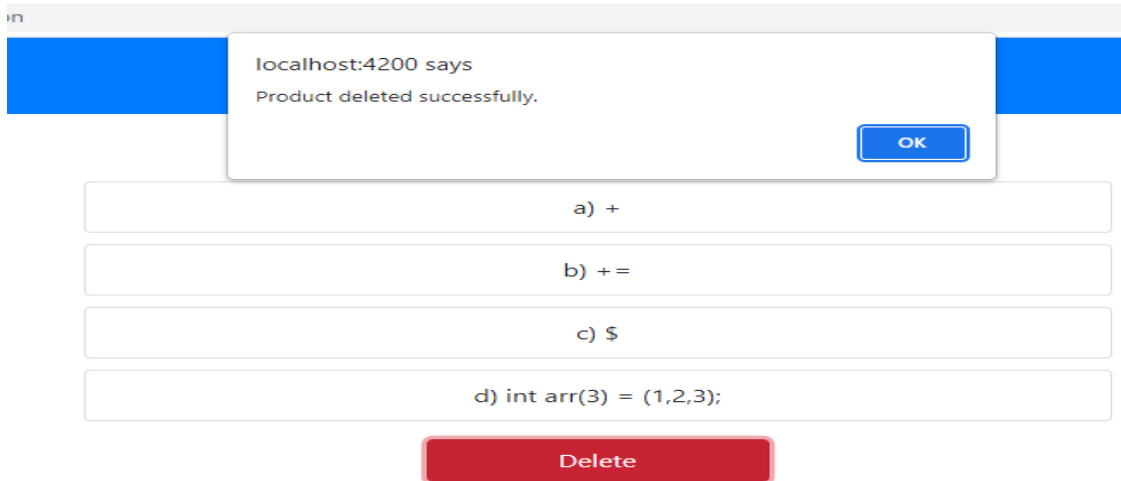
## I) Delete Question from the cart

i) If the user has added question to preview.

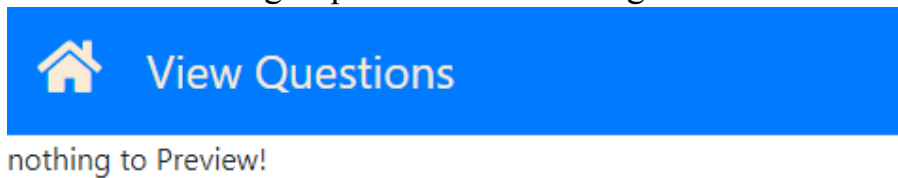
SQL Views are also known as

- a) +
- b) +=
- c) \$
- d) `int arr(3) = (1,2,3);`

Delete



ii) If the user has nothing to preview thus nothing to delete.



## J)Download Questions from Cart

i) If the user has added some questions to preview.

Which is the subset of sql commands,used to manipulate oracle database structures,including tables

- a) strinit()
- b) strnset()
- c) strset()
- d) strcset()

Delete  
Download

- c) strset()
- d) strcset()

Delete

question (2).pdf

question (2).pdf 1 / 1 66% +

How do you initialize an array in C?

- a) int arr[3]={(1,2,3);
- b) int arr(3)={1,2,3};
- c) int arr[3] = {1,2,3};
- d) ||

Delete

Which of these operators can be used to concatenate two or more String objects in Java?

- a) +
- b) +=

ii) If the user has nothing to preview thus nothing to download.

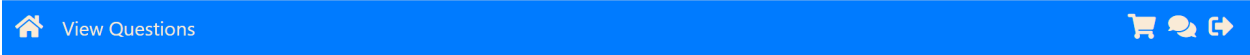




nothing to Preview!

**K)Feedback:**

i)Layout:

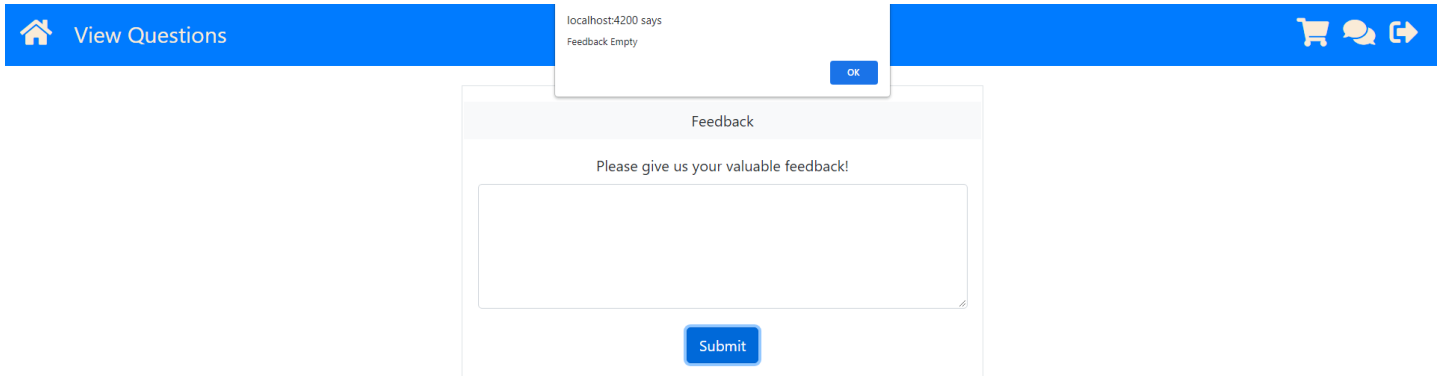


Feedback

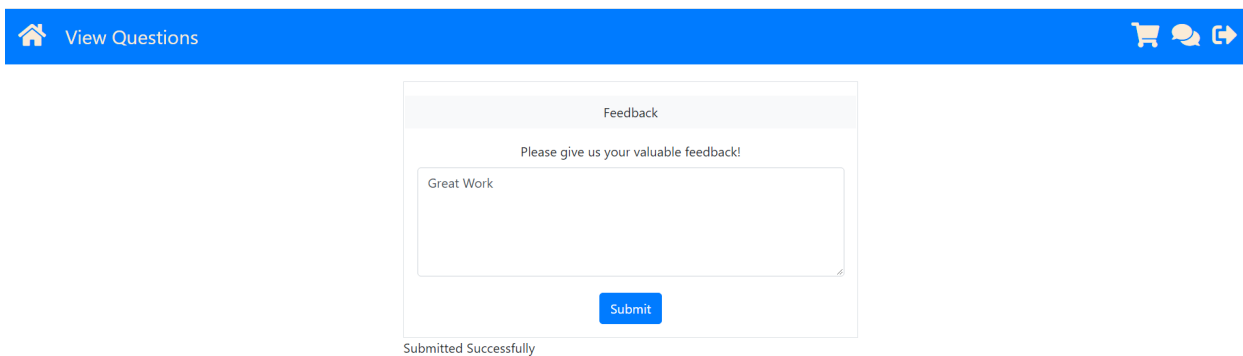
Please give us your valuable feedback!

Submit

ii)Feedback is empty:



iii)Feedback successful:



## 10. Conclusion

By the end of our internship we were successfully able to create a Web app which could generate objective questions on topic of your choice. The user could modify a certain question, delete a question from his previewable, download the set of questions, manually see all the questions from datasets and could even add a question of his/her own choice.

During the course of this Internship I learned a lot and had a experience with my fellow colleagues and mentor from training to project it was a great experience and I would surely carry the learning from the internship to my future life endeavours.

## REFERENCES

1. Hanson, David & Proebsting, Todd. (2004). A research C# compiler. *Softw., Pract. Exper.*, 34. 1211-1224. 10.1002/spe.610.
2. Freeman, Adam. (2018). Entity Framework Core in Context. 10.1007/978-1-4842-3435-8\_1.
3. Troelsen, Andrew & Japikse, Philip. (2017). Introducing ASP.NET Web API. 10.1007/978-1-4842-3018-3\_30.
4. Sharma, Aakanksha. (2018). Introduction to HTML (Hyper Text Markup Language) - A Review Paper. *International Journal of Science and Research (IJSR)*. 7. 1337-1339.
5. Genevès, Pierre & Layaida, Nabil & Quint, Vincent. (2012). On the Analysis of Cascading Style Sheets. 10.1145/2187836.2187946.
6. Durganath, Y. & Edreena, P.. (2016). Role of bootstrap in web page creation. *International Journal of Pharmacy and Technology*. 8. 21116-21122.
7. Bucea-Manea-Tonis, Radu. (2016). Angular JS – The Newest Technology in Creating Web Applications. *Annals of Spiru Haret University Economic Series*. 16. 103. 10.26458/1638.

8. Silva, Yasin & Almeida, Isadora & Queiroz, Michell. (2016). SQL: From Traditional Databases to Big Data. 413-418. 10.1145/2839509.2844560.
9. Jha, Pratibha & Khan, Rizwan. (2018). A Review Paper on DevOps: Beginning and More To Know. International Journal of Computer Applications. 180. 16-20. 10.5120/ijca2018917253.
10. Ankita Saini <https://www.geeksforgeeks.org/interesting-facts-about-c-sharp/>
11. Dave Gavigan <https://medium.com/the-startup-lab-blog/the-history-of-angular-3e36f7e828c7>
12. <https://www.edapp.com/rapid-refresh/>
13. <https://www.easytestmaker.com/Tour/Printing>
14. <https://www.typeform.com/quizzes/>
15. <https://www.classmarker.com/>
16. <https://www.proprofs.com/quiz-school/register/?demo>
17. <https://www.fyrebox.com/quiz-maker>
18. Suraj Patil [https://github.com/patil-suraj/question\\_generation](https://github.com/patil-suraj/question_generation)
19. Kristiyan Vachev <https://github.com/KristiyanVachev/Question-Generation>
20. Yuxi Xie <https://github.com/YuxiXie/SG-Deep-Question-Generation>

