Project Report

Q&A SYSTEM - TECH SOLUTION

OVERVIEW

The project titled ***"Q&A SYSTEM - TECH SOLUTION"*** is designed using Active Server Pages .NET and SQL Server 2009 which runs under .Net Framework 2015 in Microsoft Windows Operating System family.
Technical support plays a critical role in the operations of any organization. An efficient technical support ensures efficient use of computer resources and technology throughout. Thus, this must encourage use by providing excellent customer service.
Excellent customer service means that service requests are handled correctly and in a in a timely manner. Consequently, calls must be escalated to the appropriate resources when required to ensure quick resolution and more basic tasks must be handled immediately.
The Tech solution System is a set of programs which run as a software providing assistance to all the complaints, queries and services to the customers within a set of organizations. It is a web based software which has made the working procedure of an organization is much easier. Tech Solutions is a featured Q/A website for professional and enthusiast programmers.The aim of the project is to implement all the basic and necessary functionalities of StackOverFlow . For developers, by developers. Here you get answers to your toughest coding questions, share knowledge with your coworkers and find your next dream job.

GOALS

Purpose is to build an application of questions and answers.
The primary objective of the Tech Solution is to answer the query of users and provide free solutions to them.

BENEFITS

Interaction will be easier.
Users queries can be viewed by others
Less time consuming.

SYSTEM SPECIFICATION - Q&A SYSTEM

HARDWARE REQUIREMENTS

The hardware used for the development of the project is:

PROCESSOR	:	I7
RAM	:	8 GB
MONITOR	:	15” COLOR
HARD DISK	:	1 TB
MOUSE	:	3 BUTTONS

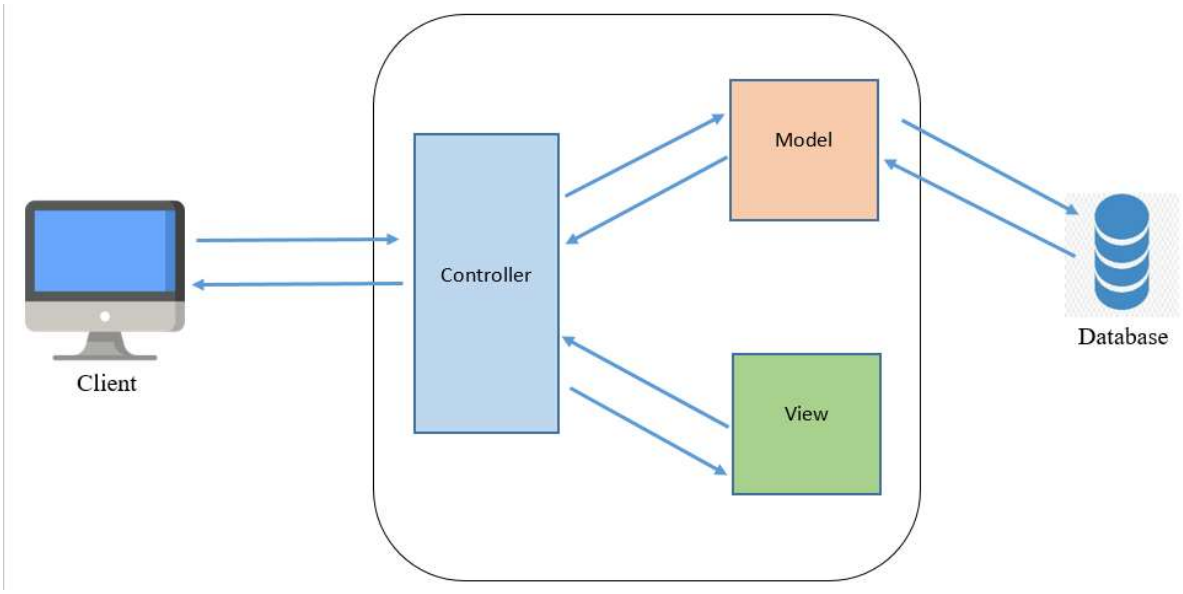
SOFTWARE REQUIREMENTS

The software used for the development of the project is:

OPERATING SYSTEM	:	Windows 10
ENVIRONMENT	:	Visual Studio .NET 2015
.NET FRAMEWORK	:	Version 4.6
LANGUAGE	:	C#.NET, ASP.NET, MVC
BACKEND	:	SQL SERVER 2009

FEATURES OF MVC

ASP.NET Core MVC is a rich framework for building web apps and APIs using the Model-View-Controller design pattern.
The Model-View-Controller (MVC) architectural pattern separates an application into three main groups of components: Models, Views, and Controllers. This pattern helps to achieve separation of concerns. Using this pattern, user requests are routed to a Controller which is responsible for working with the Model to perform user actions and/or retrieve results of queries. The Controller chooses the View to display to the user, and provides it with any Model data it requires.



MVC Pattern

This delineation of responsibilities helps you scale the application in terms of complexity because it's easier to code, debug, and test something (model, view, or controller) that has a single job. It's more difficult to update, test, and debug code that has dependencies spread across two or more of these three areas. For example, user interface logic tends to change more frequently than business logic. If presentation code and business logic are combined in a single object, an object containing business logic must be modified every time the user interface is changed. This often introduces errors and requires the retesting of business logic after every minimal user interface change.

Model Responsibilities

The Model in an MVC application represents the state of the application and any business logic or operations that should be performed by it. Business logic should be encapsulated in the model, along with any implementation logic for persisting the state of the application. Strongly-typed views typically use ViewModel types designed to contain the data to display on that view. The controller creates and populates these ViewModel instances from the model.

View Responsibilities

Views are responsible for presenting content through the user interface. They use the Razor view engine to embed .NET code in HTML markup. There should be minimal logic within views, and any logic in them should relate to presenting content. If you find the

need to perform a great deal of logic in view files in order to display data from a complex model, consider using a View Component, ViewModel, or view template to simplify the view.

Controller Responsibilities

Controllers are the components that handle user interaction, work with the model, and ultimately select a view to render. In an MVC application, the view only displays information; the controller handles and responds to user input and interaction. In the MVC pattern, the controller is the initial entry point, and is responsible for selecting which model types to work with and which view to render (hence its name - it controls how the app responds to a given request).

SYSTEM DESIGN - Q&A SYSTEM

Design is a multi-step process that focuses on data structure, software architecture, procedural details, (algorithms etc.) and interface between modules. The design process also translates the requirements into the presentation of software that can be accessed for quality before coding begins.

Computer software design changes continuously as new methods; better analysis and broader understanding evolved. Software Design is at a relatively early stage in its revolution.

Therefore, Software Design methodology lacks the depth, flexibility and quantitative nature that are normally associated with more classical engineering disciplines. However techniques for software designs do exist, criteria for design qualities are available and design notation can be applied.

INPUT DESIGN

Input design is the process of converting user-originated inputs to a computer-based format. Input design is one of the most expensive phases of the operation of a computerized system and is often the major problem of a system.
In the project, the input design is made in various web forms with various methods.
For example, in the Admin form, the empty username and password is not allowed. The username if exists in the database, the input is considered to be invalid and is not accepted.

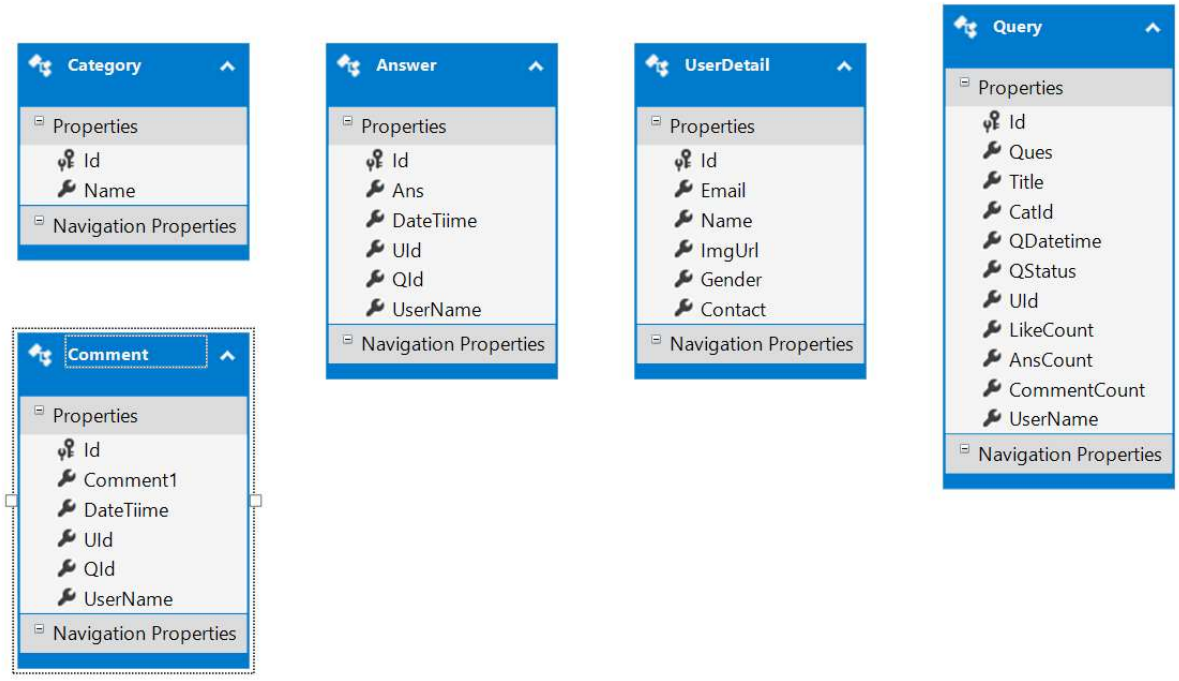
OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application.
In the project, once questions are posted, it is stored into the database. The questions are viewed and also the user who needs the details about the question can register and see the related answer which is to be posted on this site.

DATABASE DESIGN

The database design is a must for any application developed, especially more for the data store projects. Since the chatting method involves storing the message in the table and produced to the sender and receiver, proper handling of the table is a must.
In the project, the login table is designed to be unique in accepting the username and the length of the username and password should be greater than zero.

Database Table



SYSTEM IMPLEMENTATION - Q&A SYSTEM

Implementation is the most crucial stage in achieving a successful system and giving the user's confidence that the new system is workable and effective. Implementation of a modified application to replace an existing one. This type of conversation is relatively easy to handle, provided there are no major changes in the system.
Each program is tested individually at the time of development using the data and has verified that this program linked together in the way specified in the programs specification, the computer system and its environment is tested to the satisfaction of the user. The system that has been developed is accepted and proved to be satisfactory for the user. And so the system is going to be implemented very soon. A simple operating procedure is included so that the user can understand the different functions clearly and quickly.
Initially as a first step the executable form of the application is to be created and loaded in the common server machine which is accessible to all the user and the server is to be connected to a network. The final stage is to document the entire system which provides components and the operating procedures of the system.

The project contains three main modules:-

1) **User module**
2) **Admin module**
3) **Expert module**

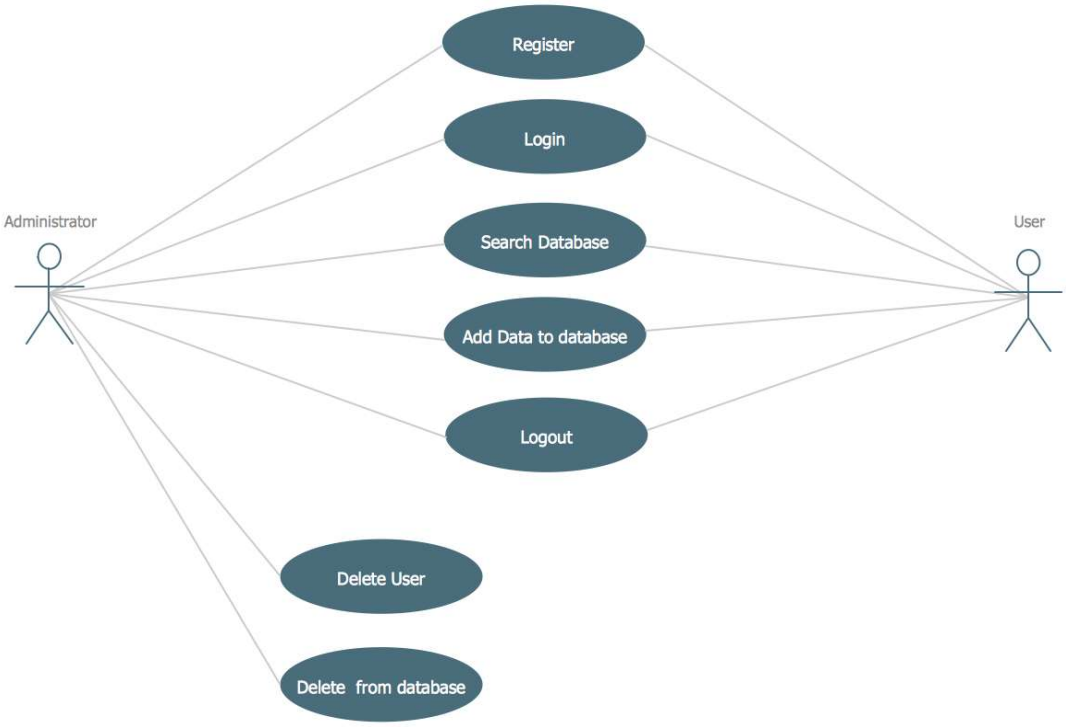
USER MODULE

In this module, the users, whether they are new or the existing ones, have to initially login into the system or help desk palace to post any query or related problems. If the user is an existing one, then he just has to login into the system by username and password which had been already provided to him at the time of registration and if it is the new user then he just has to register

ADMIN MODULE

Admin will be able to add/update/remove any user or technician manually.Admin should be able to accept/reject a client or a service provider's sign up request.
It means whenever a user or a technician signs up through the registration forms, they should not be added into the database until the admin approves.

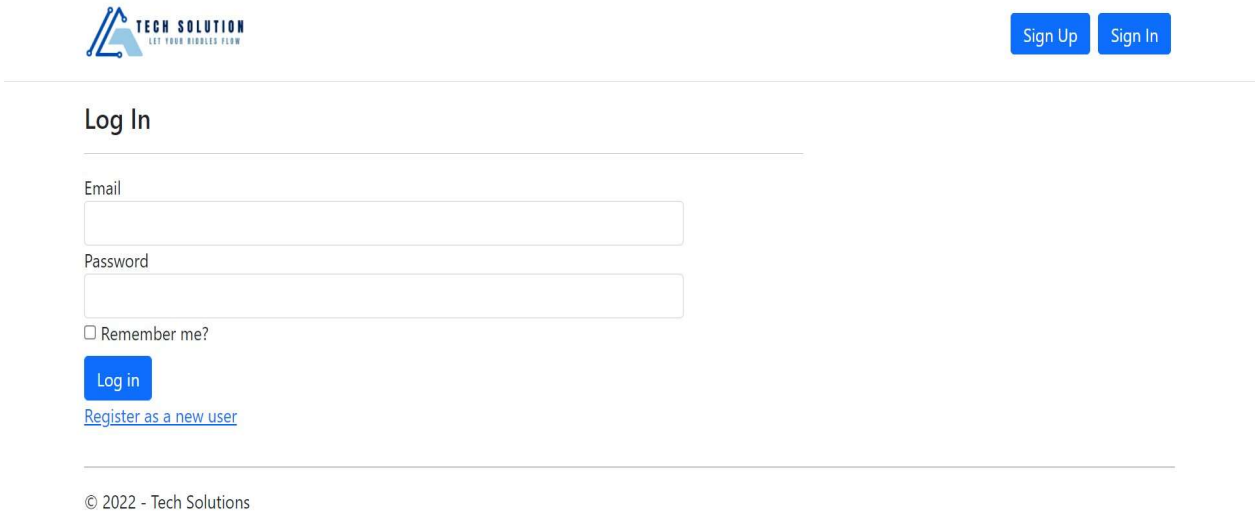
“THIS DIAGRAM SHOWS HOW USER AND ADMIN MODULES ARE CONNECTED”



EXPERT / TECHNICIANS MODULE

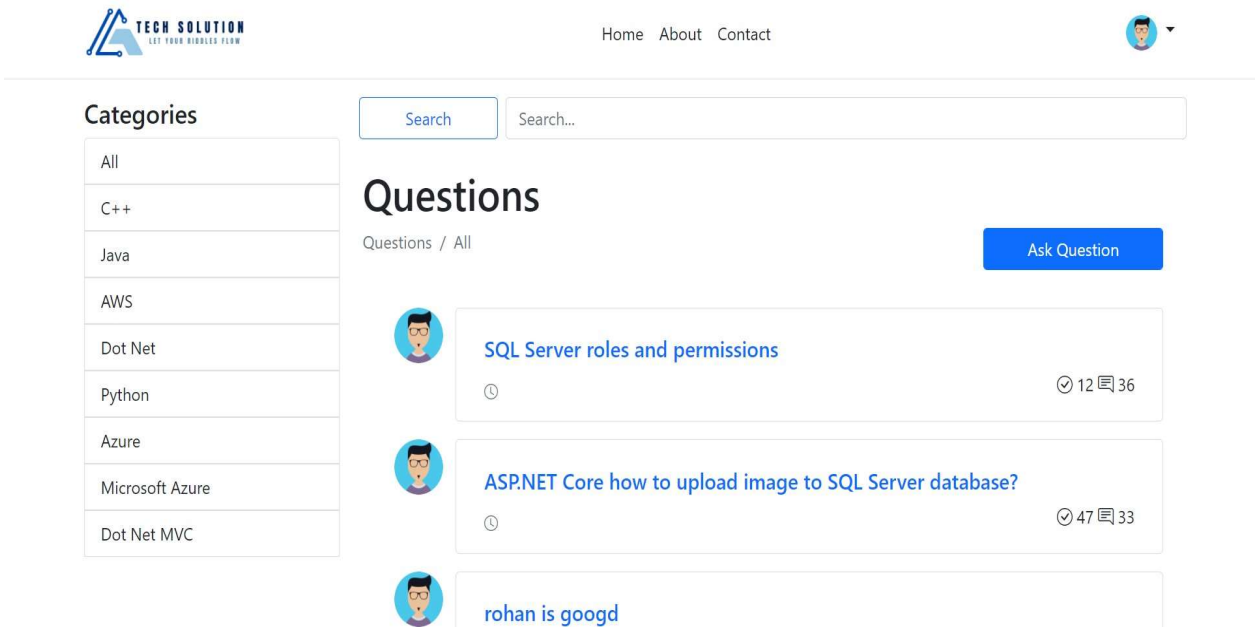
Technicians can go through and answer questions about what has been completed or done in the field. The technicians then submit this for review. The questions are created, managed, and updated by the managers.

Login Page UI Design

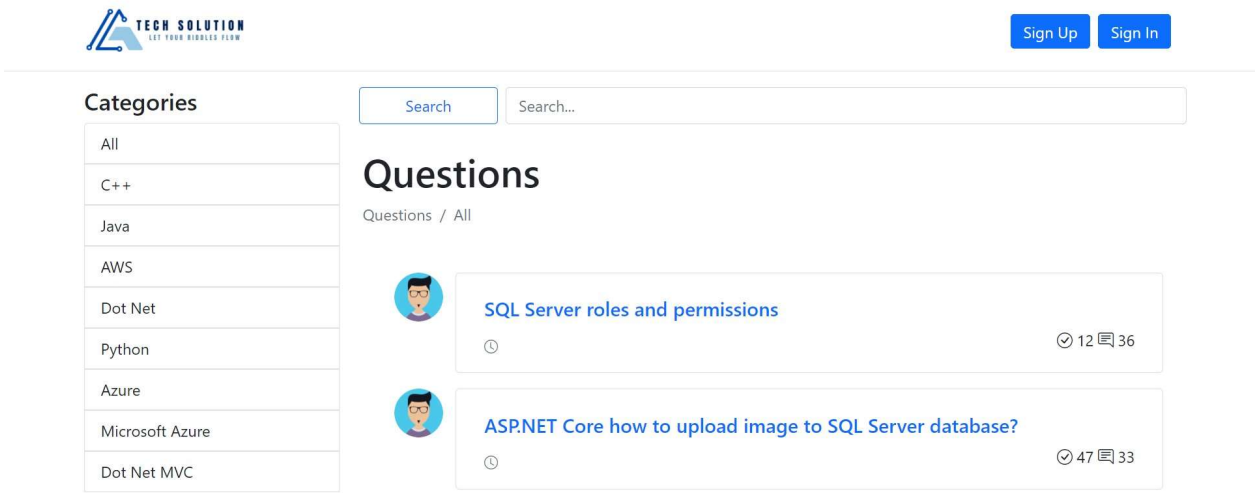


Sign Up Page UI Design

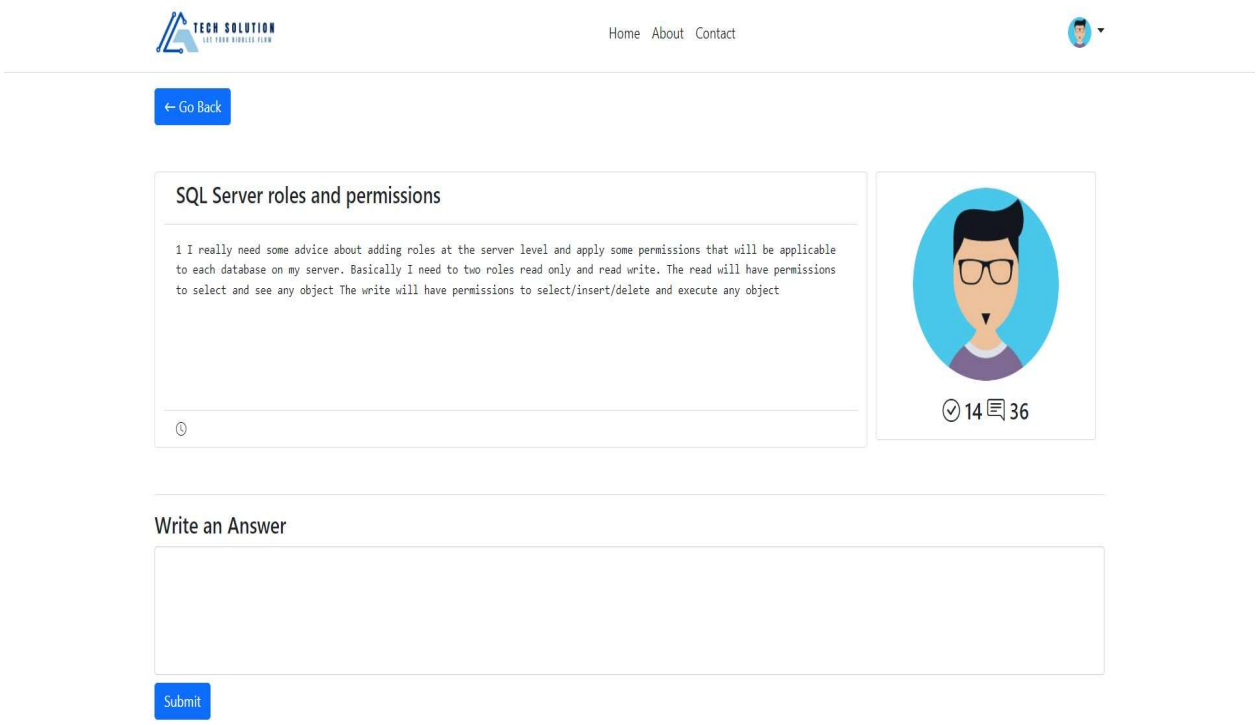
User home page View



Guest Home Page



Expert View



TEAM MEMBERS

SIVA KRISHNA
AKSHAY SOMVANSHI
HARSHITA CHOUHAN
NIHARIKA PATIDAR
GOURAV RAGHUWANSHI
SHUBHANGI PATIL
PRADHUMAN SUNER
SHUBHAM SHAH
DARSHAN CHOUDHARY
SANTOSH VERMA
NIDHI KHUSHWAH