

DATABASE

Database type chosen as SQL server type for high security and data consistency. Database technology can be any SQL server type (Postgres, MSSQL, Oracle DB etc.)

ENDPOINTS

Question Endpoints

HTTP	RETURNS (Successful)	DESCRIPTION
GET	List<QuestionViewModel>	Get All Questions
GET	QuestionViewModel	Get Question By ID
GET	List<QuestionViewModel>	Get Questions By User
GET	List<QuestionViewModel>	Get Questions By Category
POST	201 status	Insert Question
PUT	204 status	Update Question
DELETE	204 status	Delete Question

Category Endpoints

HTTP	RETURNS (Successful)	DESCRIPTION
GET	List<CategoryViewModel>	Get All Categories
GET	CategoryViewModel	Get Category By ID
POST	201 status	Insert User
PUT	204 status	Update User
DELETE	204 status	Delete User

User Endpoints

HTTP	RETURNS (Successful)	DESCRIPTION
GET	List<UserViewModel>	Get All Users
GET	UserViewModel	Get User By ID
POST	201 status	Insert User
PUT	204 status	Update User
DELETE	204 status	Delete User

Answer Endpoints

HTTP	RETURNS (Successful)	DESCRIPTION
GET	List<AnswerViewModel>	Get All Answers
GET	AnswerViewModel	Get Answer By ID
POST	201 status	Insert Answer

HTTP	RETURNS (Successful)	DESCRIPTION
PUT	204 status	Update Answer
DELETE	204 status	Delete Answer

## MODELS

User.cs

```
public class User
{
    public int Id {get; set;}
    public string FirstName {get; set;}
    public string LastName {get; set;}
    public string Email {get; set;}
    public string Password {get; set;}
    public Category FavoriteCategory {get; set;}
}
```

Question.cs

```
public class Answer
{
    public int Id {get; set;}
    public string Title {get; set;}
    public string Description {get; set;}
    public int UserId {get; set;}
    public DateTime AskDate {get; set;}
    public List<Answer> Answers {get; set;}
    public Category Category {get; set;}
}
```

Answer.cs

```
public class Answer
{
    public int Id {get; set;}
    public int QuestionId {get; set;}
    public int UserId {get; set;}
    public DateTime AnswerDate {get; set;}
    public string content {get; set;}
}
```

Categories.cs

```
public class Category
{
    public int Id {get; set;}
    public string Name {get; set;}
}
```

## AUTHENTICATION

Authenticated users will be given, secure JWT (JSON Web Token) stored in HTTP Cookie.

## USER INTERACTION LOGIC

User view and like/dislike counts can be streamed to the frontend via web sockets, so users can see other users action in real time without refreshing the page.