Pre-request every one again to join in every following groups to get further updates and if you leave groups after your TFA and stream tests you will miss your further updates for sure so just stay connected.

Dm - https://t.me/angelnndevil

Main group link - https://t.me/ad2group

Club - https://t.me/accentureclub

All Streams - https://t.me/accenturstreamgroup

Materials/Channel/dumps - https://t.me/adaccenturematerials

MVC Architecture:MVC stands for Model, View, and Controller.

Model: **Model represents the data** : A controller interacts with the model, access the data, perform the logic and pass that data to the view.

- Model represents the shape of the data.
- It is a class in C# is used to describe a model.
- Model consist of set of classes that are mapped to tables in databases.
- Model objects store data retrieved from the database.

View: **View is the User Interface.**

- View display model data to the user and also enables them to modify them.
- View in ASP.NET MVC is HTML, CSS, and JS that makes it easy to communicate with the model and the controller.

Controller: **Controller is the request handler**

➔ Def:Controller is a class that handles user requests. It retrieves data from the Model and renders view as response.

- a class contains public method known as actions ,the user request is first rcvd by controller
- the user uses the view and raises an HTTP request
- The controller processes the request and returns the appropriate view as a response.
- its action method handles incoming browser requests, retrieves necessary model data and returns appropriate responses.

What is routing? ->Routing is pattern matching system that monitors the incoming request and map it to appropriate controller and action.at runtime, the appropriate controller and action is called to display the respective view

What is action? ->whenever you send a rqst to mvc application from your browser , this rqst is rcvd by method of controller class called action. Controller can have many actions. A user request can be any of like: entering URL into the browser, clicking a link or submitting a form.

What is state management technique?

- By default web applications are stateless If they are stateless they don't retain/remember the values
- To retain this state between request/pages we use following techniques- Session, Application, Query String, Cookies

Diff btwn viewbag, viewdata, TempData ?(V B, V D, T D)

ViewBag, ViewData ,TempData

#	ViewBag	ViewData	TempData
	It's a dynamic property of the controller class	It's a key, value based dictionary	It's a key , value based dictionary
	Can transfer data from controller to view	Can transfer data from controller to view	Can transfer data from controller to view & Controller to controller
	No compiler time checking	Compiler checks the type	Compiler checks the type
	No type casting is required, easy to use for dynamic data from data source	Explicit type casting is required, but safe as compiler check is available	Explicit type casting is required, helps between controller to controller (i.e action to action data transfer)

ViewBag defining : ViewBag.SampleData =”Accenture”

Calling: @ViewBag.SampleData

ViewData defining: ViewData[“SampleData”] = “Accenture”

Calling: @ViewData[“SampleData”]

Key points

- In .Net, you cannot assign a **null** value to an **int** or any other struct. Instead, use a **Nullable<int>** or int? for short
- **navigation property**: Navigation Property represents the relationship with another entity.if an entity includes the property of another type of the entity, then it is called as **Navigation Property**.
- **Scalar Property:**It contains the actual data from the data source.Each scalar property maps to a column in the database table which stores the real data.
- **How you can navigate from one view to another view?-** By using the ActionLink method
- What is **Db set** : The DbSet class represents an entity set that can be used for create, read, update, and delete operations.
- The **DbContext** class is an integral part of Entity Framework. An instance of DbContext represents a session with the database which can be used to query and save instances of your entities to a database. -> manage database connection, configure model& relationship, querying database

➔ A **layout view** provides a consistent layout for a site. A **partial view** is a reusable component used within a View

➔ The layout view allows you to define a common site template, or example, an application UI may contain a header, left menu bar, right bar, and footer section that remains the same on every page. Only the center section changes dynamically,

➔ How to transfr data frm **Controller to view**. (ans. Viewdata)

➔ A **COMMIT statement** is used to save the changes on the current transaction is permanent. A **Rollback statement** is used to undo all the changes made on the current transaction.

➔ **App_start** -> contains files that execute when application starts ex: route config.cs

➔ **App_Data** -> this folder is used to store files-based database eg. Xml files

➔ **Global.Asax** -> it contains events like app_start and session_start

➔ **Can two controllers redirect to same view** -> yes, by using RedirectToAction() Method

➔ **Partial view can be rendered using html helper methods like** -> Partial() or RenderPartial() or RenderAction()

➔ Action Result is actually a data type. When it is used with action method, it is called return type

➔ Action result subtypes

- **ViewResult** - Renders a specifed view to the response stream
- **PartialViewResult** - Renders a specifed partial view to the response stream
- **EmptyResult** - An empty response is returned
- **RedirectResult** - Performs an HTTP redirection to a specifed URL
- **RedirectToRouteResult** - Performs an HTTP redirection to a URL that is determined by the routing engine, based on given route data
- **JsonResult** - Serializes a given ViewData object to JSON format
- **JavaScriptResult** - Returns a piece of JavaScript code that can be executed on the client
- **ContentResult** - Writes content to the response stream without requiring a view
- **FileContentResult** - Returns a file to the client
- **FileStreamResult** - Returns a file to the client, which is provided by a Stream
- **FilePathResult** - Returns a file to the client

➔ Viewstart.cshtml->it is used to set common layout across all the views

Entity Framework

Entity Framework: The Entity Framework is a development platform that provides a layer of abstraction over the data sources used by applications (data source can be xml files,relational DB, collections)

- entity framework is a tool we used to access database & is responsible for opening a connection to the DB, reading of data, And also maps the data to the objects
- It is a object relational mapper(ORM) which we used to map the objects of our application with the relational database
- Entity framework provides us a class called a dB context which is a gateway to our database
- A dB context can have one/more dB sets(these Dbsets represents table in our database)
- We use linq queries to code dbset and EF translates linq to sql at runtimeand makes changes in db

ORM(object relational mapping) : An ORM provides a framework for developers to connect applications to the data sources without making changes to data model.

- To understand EF it's important to know ORM as an application should be visualized as two layers data model and object model
- Data model can be relational db, flat files , collections and what ever the data might be it is considered as an object
- Consider an shopping application which uses a relational db to store the info abt its product and its categories and these both tables are linked using foreign key
- But in todays oop world we don't have table, column. So how do we define them in .net?
- The answer is to use classes which relates business entity(products, categories) and use relationships in oops like inheritance, dependency
- The object model contains classes with variables to store data and relationship between classes must be represented using inheritance this concept is called an ORM

➔ We have 3 types of **EF workflows**

- **Database First**: we design our own tables,as this approach sets up an object model on an existing database
- Ef generates domain classes, it uses the tables and columns from database to generate the classes for the business model
- **Code first**: We create our domain classes, EF generates database tables
- Ef provides a designer tool which generates the database schema used to create tables
- **Model first**: we create a Uml diagrams, Ef generates domain classes and database

➔ **Entity Data model**: it is the core of the entity framework. The EDM abstracts the data model and exposes conceptual schema using layered approach.

➔ The layers are:

- **Conceptual layer**: the main purpose is to present oop model of data source for developers. The conceptual model contains the modal classes and their relationships.
- **Mapping layer**: This layer defines the mapping btwn the conceptual and storage layer
- **Storage Layer**: This layer contains the components of the data source used by application in this case it would be a relational db which contains (tables, views,stored procedures)

SQL

➔ **Views**: Views in SQL are considered as a virtual table. A view also contains rows and columns.-> they don't take any physical storage

- View is the result set of a stored query(base class)
- A view can either have specific rows based on certain condition or all the rows of a table.
- Syntax: create view v1 as select id from student.
- If we make any changes in base table then values in view will also be changed(updatable views)
- Read only views are those which shutdowns insert, update actions on views

➔ **Joins**: The SQL JOIN clause takes records from two or more tables in a database and combines it together.

➔
```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;
```

(INNER) JOIN: Returns records that have matching values in both tables
LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table

➔ **Stored procedure**: it is a prepared sql code which can be saved and reused again and again

- You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.
- Create a SP: Create Procedure proced_name as sql_stmt go; to execute: EXEC procd_name
- Create a sp (parameter): Create procedure @City nvarchar(30) as sql_stmt go;
- The main purpose of stored procedures to hide direct SQL queries from the code and improve performance of database operations such as select, update, and delete data.

➔ **Constraints**

• **NOT NULL**:NOT NULL Constraint forces A Column To NOT Accept NULL Values, This enforces a field to always contain a value,
• **UNIQUE**: all the values in the column must be unique. the values in any row of a column must not be repeated, doesn't allow any duplicate values.
• **PRIMARY KEY**: A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARYKEY in another table. The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.
• **FOREIGN KEY**: A Foreign key is a field which can uniquely identify each row in another table. And this constraint is used to specify a field as Foreign key.
• **CHECK**: This constraint helps to validate the values of a column to meet a particular condition.
• **DEFAULT**: This constraint specifies a default value for the column when no value is specified by the user.

Key Point:

- ➔ The key difference between **varchar and nvarchar** is the way they are stored, varchar is stored as regular 8-bit data(1 byte per character) and **nvarchar stores data at 2 bytes per character**
- ➔ **An identity column is a column in a database table that is made up of values generated by the database.**
- ➔ **Sql delete** : It removes rows one at a time. It deletes structure also
  Sql truncate : It removes all rows in a table by deallocating the pages that are used to store the table data . It preserves the table structure
- ➔ **can u delete record from table using view? ->** If you have created a View in SQL which is based on a single table – the DML operations you perform on the view are automatically propagated to the base table.
- ➔ **LINQ to Entities :** that enables developers to write queries against the Entity Framework conceptual model using c#
- ➔ **Linq to SQL**: LINQ to SQL allow you to query and modify SQL Server database by using LINQ syntax.
- ➔ **Can unique column have multiple null values?** -> UNIQUE constraint allows multiple NULLs. But in the SQL Server, it allows only one NULL value. With the UNIQUE constraint, you cannot insert multiple NULLs
- ➔ **Validation controls**

RequiredFieldValidation	Makes an input control a required field
CompareValidator	Compares the value of one input control to the value of another input control or to a fixed value
RangeValidator	Checks that the user enters a value that falls between two values
RegularExpressionValidator	Ensures that the value of an input control matches a specified pattern
CustomValidator	Allows you to write a method to handle the validation of the value entered
ValidationSummary	Displays a report of all validation errors occurred in a Web page

http get ,post, delete

Http method	Usage
GET	To retrieve the information from the server. Parameters will be appended in the query string.
POST	To create a new resource.
PUT	To update an existing resource.
HEAD	Identical to GET except that server do not return the message body.
OPTIONS	It represents a request for information about the communication options supported by the web server.
DELETE	To delete an existing resource.
PATCH	To full or partial update the resource.

Explain Scenario:

- ➔ Acc corp Is offering insurance scheme to its employees which includes employee,spouse,and children as additional benefit they decided to offer its employees nominate there parents/in-laws as their dependents for insurance coverage
  - Employee can increase there insurance coverage that would be deducted from salary
  - Employee can nominate many relations as dependents but can have only single coverage amount
    Application
  - Login with their corporation id
  - View/ increase insurance coverage amount
  - View/add dependent information
    Contollers- 3 (acc corporation ins portal,view dependents, view coverage)
    Model- 3 login creds validation,add dependents, increase coverage
    View- 3 view dependents, view coverages,acc corporation ins portal
    Actions- add, update,delete

- Tables- Employee(emp id, emp name,email,pass)
  Nominee(nominee id, emp id, nom name, relation, dob)
  Employee Insurance(id, emp id, coverage amt, validity end date)

1.Explain routing and types of routing?

A.Routing is a process of mapping the brower request to the controller action and retun response back.each mvc application has default routung for the default homecontroller.and also we can set the custom routing for newly

created controller.

types- convention based routinh(Maproute), attribute based routing(route).

2.How to find primary key in Entity framework?

A.EF core find method finds a record with the given primary key values.if the entity is already in the context,Then the find method returns it.if not the query sents to the database to return the entity.

3.What is DBcontext classes?

A.DBcontext is an important class in entity framework API.it is bridge between domain or entity classes and database.DBcontext is the primary class that is responsible fot interacting with the database.

4.what is DBset and Entity set?

A.DBset represents the collection of all the entities in the context.or that can be queried from the database.

DBset object are created from the DBcontext.

Entity set: entity set is a logical container for instance of an entity type and instance of any type drived from that entity type.

5.use of find in Entity framework?

A.The Find method on DBset uses the primary key value to attempt to find an entity tracked by the context.

if not the query sents to the database to return the entity.

6.Scalar and navigation property in entity framework?

A.Scalar property: A property of an entity that maps a single field in the storage model is called scalar property.

example: studentId, studentname(int, string)

navigation property: It is used to navigate through the relations in data.It allows us to navigate from one entity to connected entity.

7.What is LINQ to entities?

A.LINQ to entities provides(LINq) langauge intigrated query supports that enable developers to write query against the entity framework conceptual model using visual c# etc.

right-click on the Shared folder -> select Add -> click on New Item...

Controller

The Controller in MVC architecture handles any incoming URL request.

Controller class contains public methods called Action methods. Controller and its action method handles incoming browser requests, retrieves necessary model data and returns appropriate responses.

In the Visual Studio, right click on the Controller folder -> select Add -> click on Controller..

Scaffolding is an automatic code generation framework for ASP.NET web applications. Scaffolding reduces the time taken to develop a controller, view, etc. in the MVC framework. You can develop a customized scaffolding template using T4 templates as per your architecture and coding standards.

All the public methods of the Controller class are called Action methods. They are like any other normal methods with the following restrictions:

Action method must be public. It cannot be private or protected

Action method cannot be overloaded

Action method cannot be a static method.

Every controller can have a default action method as per the configured route in the RouteConfig class.

The model classes represents domain-specific data and business logic in the MVC application. It represents the shape of the data as public properties and business logic as methods.

folder, select Add -> and click on Class... It will open the Add New Item

Razor:(RZ)

special syntax in view

Razor allows you to write a mix of HTML and server-side code using C# or Visual Basic. Razor view with visual basic syntax has `.vbhtml` file extension and C# syntax has `.cshtml` file extension.

Razor syntax has the following Characteristics:

- **Compact**: Razor syntax is compact, enabling you to minimize the number of characters and keystrokes required to write code.
- **Easy to Learn**: Razor syntax is easy to learn where you can use your familiar language C# or Visual Basic.
- **Intellisense**: Razor syntax supports statement completion within Visual Studio.

ViewBag

The ViewBag in ASP.NET MVC is used to transfer temporary data (which is not included in the model) from the controller to the view.

Action filter executes before and after an action method executes. Action filter attributes can be applied to an individual action method or to a controller. When an action filter is applied to a controller, it will be applied to all the controller's action methods.

A partial view is a reusable portion of a web page. It is .cshtml or .vbhtml file that contains HTML code. It can be used in one or more Views or Layout Views. You can use the same partial view at multiple places and eliminates the redundant code.

Partial View is used to display repeated summering information (like user control in ASP.NET Web Form)

Layout View is designed for the application that has different section for each block and to display same layout to entire web application (Like Master Page in ASP.Net Web Form)

Partial view just render itself without calling his paranet.

you can call partial view by jQuery/Javascript means partial view can be calling use Ajax or Ajah.

Suppose you are developing an ASP.NET web application and you want to maintain a consistent look and feel across all the pages within you web application. You then have two options, the first one is to design the head, body and footer sections on each page. In this approach you need to write more code on each page so ASP.NET 2.0 introduced "Master Pages" that helps enable this when using .aspx based pages or templates. It is your second option. Razor also supports this concept with a feature called "layouts" that allow you to define a common site template, and then inherit its look and feel across all the views/pages on your web application.

1. difference between primary key and unique key and foreign key
   https://www.c-sharpcorner.com/blogs/difference-between-primary-key-unique-key-and-foreign-key1

   A primary key is used to ensure data in the specific column is unique
   A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It uniquely identifies a record in the relational database table. ... Whereas more than one foreign key are **allowed** in a table.
   A unique key is **a set of one or more than one fields/columns of a table that uniquely identify a record in a database table**

   Null
   Duplicates
   Not Null : tells column not take null values

3.foreign keys and primary keys are not strictly necessary for join queries

SQL JOINS are used **to retrieve data from multiple tables**

https://www.w3schools.com/sql/sql_join.asp

ON keyword is used in joins

Inner join

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

Left Join

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

Right Join

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

Full outer Join

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```

Self Join

A self join **allows you to join a table to itself**. It is useful for querying hierarchical data or comparing rows within the same table

The best example of self join in the real world is **when we have a table with Employee data and each row contains information about employee and his/her manager**. You can use self join in this scenario and retrieve relevant information

```
SELECT column_name(s)
FROM table1 T1, table1 T2
WHERE condition;
```

Cross Join

when you wish to create a combination of every row from two tables.

```
SELECT column_name(s)
FROM table1 T1, table1 T2
WHERE condition;
```

4. SQL constraints are **a set of rules implemented on tables in relational databases to dictate what data can be inserted, updated or deleted in its tables**.

9 COMMIT permanently saves the changes made by current transaction. ROLLBACK undo the changes made by current transaction. Transaction can not undo changes after COMMIT execution

MVC

https://www.c-sharpcorner.com/article/creating-your-first-mvc-5-project-understanding-its-folder-structure/

View in MVC is **a user interface**. View display model data to the user and also enables them to modify them. View in ASP.NET MVC is HTML, CSS, and some special syntax (Razor syntax) that makes it easy to communicate with the model and the controller.

Entity Frame work

Entity framework is **an ORM (Object Relational Mapping) tool**. Object Relational Mapping (ORM) is a technique of accessing a relational database; . i.e., whatever has tables and store procedure these things we interact with database in class, method and property of the object format

Form collection:(Fc)

Form collection is used to retrieve input elements from the controller action method. Form collection class automatically receives the data form value in controller methods in the form of key/value pair. Key and value pairs are accessed using the key name and index value

MODEl BINDING: MB

The model binding refers to converting the HTTP request data (from the query string or form collection) to an action method parameters.

HTML HELPER: HT help: html hrelper is a special class in razor syntax. Instead of writing all the html tags manully we simply use html helper. @ html is the html helper.

SQL Server

1. **Difference between unique key and primary key?**
   Ans:

Primary Key	Unique Key
Unique identifier for rows of a table	Unique identifier for rows of a table when primary key is not present
Cannot be NULL	Can be NULL
Only one primary key can be present in a table	Multiple Unique Keys can be present in a table

2. **Can unique column have multiple null values?**
   Ans: No (because the second null will be duplicated)

3. **What is join and its types?**
   ---- It is used to combine records from two or more tables in a database. A JOIN is a means for combining fields from two tables by using values common to each.
   - Inner Join
   - Left Outer Join
   - Right Outer Join
   - Full Outer Join
   - Cross Join
   - Self-Join

4. **Check constraint and Default constraints**
   Check Constraint: Used to ensure that only a specific range or list of values can be stored in a column(age>18)
   Default Constraint: Used to store the default values if no value is specified (current date)

5. **Define constraints and its types**
   SQL constraints are used to specify rules for the data in a table. **Constraints** are used to limit the type of data that can go into a table
   - Not null
   - Primary Key
   - Foreign Key
   - Unique Key
   - Check
   - Default

6. **Stored procedures and parameters**
   - A stored procedure is a prepared SQL code that you can save, so the code can be reused repeatedly.

- So, if you have an SQL query that you write repeatedly, save it as a stored procedure, and then just call it to execute it.
- You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.

7. What is Identity? Parameters of identity? How Identity property works?

- Identity column of a table is a column whose value increases automatically.
- The value in an identity column is auto generated by the SQL server.
- Only one identity column is possible for a table.
- A user generally cannot insert a value into an identity column.
- Identity column can be used to uniquely identify the rows in the table.
- IDENTITY [(seed, increment)]

8. Difference between nvarchar and varchar

- The key difference between varchar and nvarchar is the way they are stored; varchar is stored as regular 8-bit data (1 byte per character) and nvarchar stores data at 2 bytes per character.
- Due to this reason, nvarchar can hold up to 4000 characters and it takes double the space as SQL varchar.

9. Difference between commit and rollback

COMMIT	ROLLBACK
Commit is a Transaction control language which is used to permanently save the changes done in the transaction in tables/database.The database cannot regain its previous state after the execution of it.	**Rollback is also a transaction control language which is used to undo the transactions that have not been saved in database. The command is only used to undo the changes since the last COMMITTEE.**

10. View in SQL

- Views in SQL are kind of virtual tables.
- A view also has rows and columns as they are in a real table in the database.
- We can create a view by selecting fields from one or more tables present in the database.
- A View can either have all the rows of a table or specific rows based on certain condition.

11. Can you delete a record form table using view?

Yes

12. Can we apply check constraints to int and varchar?

Yes

13. What is @@Identity?

We use system function @@IDENTITY to return the maximum used IDENTITY value in a table for the IDENTITY column under the current session. Once we insert a row in a table, the @@IDENTITY function column gives the IDENTITY value generated by the statement.

14. What is Global variable?

Global variables are predefined system functions. Their names begin with @@ prefix. The server maintains the value of these variables. Global variables return various pieces of information about the current user environment for SQL server.
Eg.,@@CONNECTIONS, @@ERROR,@@IDENTITY

15. Difference between where and having clause , between and like

WHERE CLAUSE	HAVING CLAUSE
Where clause is used to filter the records from the table based on the specified condition.	**Having clause is used to filter the record from the groups based on the specified condition.**
Where clause can be used without group by clause	**Having clause cannot be used without Groupby clause.**
Where clause implements in row operations.	**Having clause implements in column operations.**
Where Clause can be used with SELECT, UPDATE, DELETE	**Having clause can only be used with SELECT statement.**

LIKE OPERATOR	BETWEEN OPERATOR
LIKE operator is used to extract records where a particular pattern is present.	**BETWEEN Operator is used to select values within a given range.**
Eg, WHERE Employee Name LIKE 'e%' ---- Finds values that start with "e".	**Eg., To extract only those records where the age of the person is between 20 and 25.,we use BETWEEN query in SQL**

16. How do you get the last record from the table?

Select * from table order by ID desc limit 1

(Or)

Select top 1 * from table_name order by 1 desc

MVC and Entity Framework

1. **Explain EDM (Entity Data Model)?**

   EDM abstracts (hides) the data model and exposes a conceptual schema of the same to the developers using a layered approach.

   That layers exposed are:

   - Conceptual layer – entity classes and their relationships
   - Mapping Layer – maps the relationship between the conceptual and storage layer
   - Storage Layer – tables, views and stored procedures

2. **Action Filters in MVC**

   Action filter executes before and after an action method executes. Action filter attributes can be applied to an individual action method or to a controller. (Output Cache, Handle error, Authorize)

3. **Action Result and its types**

   Controller is responsible for returning response to the received request, which is indicated as action method's return type. These return type of action methods are called action results.

   - Action Result (base class)
   - View Result – renders. cshtml view page
   - Partial View Result – renders a partial view that can be called in regular view or layout view
   - Redirect Result - redirects to given URL
   - Redirect To Action Result/ Redirect To Route Result – takes to another action
   - Json Result – gives serialized json data
   - File Path Result – download a file from server
   - Content Result - provides user defined content
   - Java Script result – gives JavaScript that can be executed on the client browser
   - Empty result – used when action method must return null or its return type is void

4. **What is state management and its type?**

   By default, web applications are stateless, they don't remember the values stored between 2 multiple requests of same page or 2 different pages. So, to retain the state, we use state management techniques: Old (Session, Application, Query String, Cookies)

   - View bag
   - View Data
   - Temp Data

5. Explain ViewBag, ViewData and TempData.

ViewBag, ViewData, and TempData all are objects in ASP.NET MVC and these are used to pass the data in various scenarios. The following are the scenarios where we can use these objects.

1. Pass the data from Controller to View.
2. Pass the data from one action to another action in the same Controller.
3. Pass the data in between Controllers.
4. Pass the data between consecutive requests.

ViewBag

ViewBag is a dynamic object to pass the data from Controller to View. And, this will pass the data as a property of object ViewBag. And we have no need to typecast to read the data or for null checking. The scope of ViewBag is permitted to the current request and the value of ViewBag will become null while redirecting.

```
1. Public ActionResult Index()
2. {
3.     ViewBag.Title = “Welcome”;
4.     return View();
5. }
```

View

```
1. <h2>@ViewBag.Title</h2>
```

ViewData

ViewData is a dictionary object to pass the data from Controller to View where data is passed in the form of key-value pair. And typecasting is required to read the data in View if the data is complex and we need to ensure null check to avoid null exceptions. The scope of ViewData is like ViewBag and it is restricted to the current request and the value of ViewData will become null while redirecting.

Controller:

```
1. Public ActionResult Index()
2. {
3.     ViewData[”Title”] = “Welcome”;
4.     return View();
5. }
```

View

```
1. <h2>@ViewData[“Title”]</h2>
```

TempData

TempData is a dictionary object to pass the data from one action to other action in the same Controller or different Controllers. Usually, TempData object will be stored in a session object. Tempdata is also

required to typecast and for null checking before reading data from it. TempData scope is limited to the next request and if we want Tempdata to be available even further, we should use Keep and peek.

```
1.  Public ActionResult Index()
2.  {
3.      TempData[”Data”] = “I am from Index action”;
4.      return View();
5.  }
6.
7.  Public string Get()
8.  {
9.      return TempData[”Data”] ;
10. }
```

To summarize, ViewBag and ViewData are used to pass the data from Controller action to View and TempData is used to pass the data from action to another action or one Controller to another Controller.

VIEW DATA	VIEW BAG	TEMP DATA
Used to pass data from controller to view	**Used to pass data from controller to respective view**	**Used to pass data from current request to the next request**
It is available for current request only	**It is available for current request only.**	
It is derived from ViewDataDictionary class		**It is derived from TempDataDictionary class.**
Requires Typecasting for complex data.	**Doesn’t requires typecasting for complex data type.**	**Requires typecasting for complex data**
If redirection occurs,its values become null	**If redirection occurs,then its values becomes null.**	**It helps to maintain data when we move from one action to another.**

6. **What is validation and how can we implement in MVC?**

   Used to validate the data filled in the web page. The attributes that perform the validation are

   - [Required]
   - [Display (Name = “Email Id”)]
   - [Datatype (Datatype. Password)]
   - [Range ()]
   - [Regular Expression ()]
   - [Phone]

7. Action name attribute in MVC

ActionName attribute is an action selector which is used for a different name of the action method. We use ActionName attribute when we want that action method to be called with a different name instead of the actual name of the method

8. Explain routing (Custom routing). Types of routing

ASP.net introduced routing to eliminate the need mapping each URL to a physical file. Routing enables us to define a URL pattern that maps to the request handler. In ASP. Net webform application, request handler is. aspx file and in MVC , it is the controller class and Action method.

Types of routing:

1.Convention based routing - To define this routing we call MapRoute method

2.Atrribute based routing- to define this type of routing, we specify the Route attribute in action method of the controller.

9. Tell about layer in Entity Framework.

The layers exposed are:

- Conceptual layer – entity classes and their relationships
- Mapping Layer – maps the relationship between the conceptual and storage layer
- Storage Layer – tables, views and stored procedures

10. How to find primary key in Entity Framework?

11. What is dbcontext classes?

DbContext is primary class that is responsible for interacting with the database. It is a bridge between your domain or entity classes and the database.
Tasks done by DbContext:
1.Querying- converts LINQ to Entities queries to SQL query and sends them to the database.
2.Change tracking- keeps tracks of changes that occured on the entities after querying from the database.

12. What is DB set and entity set? Uses?

DbSet represents the set of entities.
In a database, a group of similar entities is called Entity Set.
DbSet enables users to perform various operations like add, remove, update on entity set.
The DbSet is responsible for performing all the basic CRUD operations on each of the entity.

13. Use of Find in Entity Framework.

Find() - finds an entity with the given primary key values. If an entity with the given primary key is being tracked by the context, then it is returned immediately without making a request to the database.

14. Scalar and Navigational property in Entity framework

Scaler property are properties whose actual values are contained in the entity.
Eg., Student entity has scaler properties. Eg., studentid, studentname,.These correspond with the Student table.

Navigational property is used to navigate through relations in data. It allows you to navigate from one entity to a "connected" Entity.Eg., It is equivalent to foreign key relationship in a database

15. Layout.cshtml

Layout.cshtml- contains the common UI portions so that we don't have to write the same code in every page.
Layout view allows you to define a common site template which can be inherited in multiple pages of an application. Layout view enhances maintenance.

16. Difference between http get and post

HTTP GET	HTTP POST
In GET method,values are visible in the URL.	**In POST method,values are not visible in the URL.**
GET has a limitation on the length of the values,generally 255 characters.	**POST has no limitation on the length of the values.**
GET method supports only string data types.	**POST supports different data types such as string,numeric,binary**
GET request can be bookmarked.	**POST request cannot be bookmarked.**
GET parameters remain in web browser history.	**POST parameters are not saved in web browser history.**

17. What is LINQ to entities

LINQ to Entity means writing the LINQ queries over the entity framework object. By using these entities we can perform any operation like insert, delete, update, etc

18. Define data model, object model

Data models define how the logical structure of a database is modelled. Data models define how data is connected to each other and how they are processed and stored inside the system.

The object model is made up of hierarchy of objects. The server object is the top level object and all instance class object reside under the server object.

19. If we want to use same view throughout the application, what do we use?

Viewstart.cshtml

20. Rules of razor view

1.Variables and functions with inline property must start with @
2.Variables are declared with the var keyword
3.Razor block having C# code always enclosed with in @{... }
4.Code statements always end with semicolon
5.C# file will always have extension "cshtml"

21. Brief of entity framework

Entity Framework is an open-source ORM framework for .NET applications supported by Microsoft. It enables developers to work with data using objects of domain specific classes without focusing on the underlying database tables and columns where this data is stored.

22. How do we use handle error in MVC?

HandleError Attribute

23. Stored procedures and parameters

A stored procedure is a prepared SQL code that we can save, so the code can be reused over and over again.So if we have an SQL query that we have to write over and over again, we save it as a stored procedure, and then just call it to execute it.

Stored Procedure Syntax:

```
CREATE PROCEDURE procedure_name

AS

 sql_statement

GO;
```

Execute the stored procedure above as follows:

```
EXEC SelectAllCustomers;
```

Stored Procedure With One Parameter

```
CREATE PROCEDURE SelectAllCustomers @City nvarchar(30)
AS
SELECT * FROM Customers WHERE City = @City
GO;
```

Stored Procedure With Multiple Parameters

```
CREATE PROCEDURE SelectAllCustomers @City nvarchar(30), @PostalCode nvarchar(10)
AS
SELECT * FROM Customers WHERE City = @City AND PostalCode = @PostalCode
GO;
```

24. Partial vs render partial

Partial generates the HTML from the View and returns it to the View to be incorporated into the page.

Render Partial doesn't return anything and instead, adds it's HTML directly to the Response objects output

25. How to search value using pk
26. DB first approach

Datebase First Approach creates the Entity Framework from an existing database. It creates model codes from the database. The database in the project and those classes become the link between the database and controller

27. What is controller, base class of controller?

A controller is responsible for controlling the way that a user interacts with an mvc application.

Controller Base class represents the base class of all mvc controllers.
It has namespace System. Web. Mvc

28. How do you code inside View?
29. Where do you find route table?
Ans: routeconfig.cs file inside app_start folder

30. How to use stored procedures in MVC?

If we want to work with the Stored Procedure then we need to use the Code First approach

31. What does precompile means in stored procedure

With stored procedures, the subsequent execution of
the same procedure will by-pass the Five steps process and go straight
to the Execution plan. This is called as Pre-Compiled plan.

32. Difference between layout view and partial view

LAYOUT VIEW	PARTIAL VIEW
A layout view provides a consistent layout for a site	A partial view is a reusable component used within a view

33. What will router do?

A router is a networking device that forwards data packets between computer networks.

34. Why do we need to perform model translation in MVC?

Ans: MVC gives you a starting place to translate your ideas into code, and it also makes coming back to your code easier, since you will be able to identify which code does what. In addition, the organizational standard MVC promotes makes it easy for other developers to understand your code.

35. How to modify edmx if changes are made in the database?

Update model

36. What are helper methods?

- HTML Helpers are methods that return a string.
- Helper class can create HTML controls programmatically. HTML Helpers are used in View to render HTML content.

The HtmlHelper class is designed to generate UI

There are different types of helper methods.

Createinputs − Creates inputs for text boxes and buttons.

Createlinks − Creates links that are based on information from the routing tables.

Createforms − Create form tags that can post back to our action, or to post back to an action on a different controller.

HOW TO NAVIGATE FROM ONE VIEW TO ANOTHER?

ACTIONLINK METHOD WILL HELP US TO NAVIGATE FROM ONE VIEW TO ANOTHER.

37. Use of OUT in stored procedures (output parameter)

Output parameter is a parameter whose value is passed out of the stored procedure, back to the calling SQL block. An OUT parameter must be a variable, not a constant. We cannot assign a default value to an OUT parameter outside of the module's body. In other words, an OUT parameter behaves like an uninitialized variable.

Input parameter is a parameter whose value is passed into a stored procedure/function module. The value of an IN parameter is a constant; it can't be changed or reassigned within the module.

An **input/output parameter** is a parameter that functions as an IN or an OUT parameter or both. The value of the IN/OUT parameter is passed into the stored procedure/function and a new value can be assigned to the parameter and passed out of the module. An IN/OUT parameter must be a variable, not a constant. However, it can be found on both sides of an assignment. In other words, an IN/OUT parameter behaves like an initialized variable.

38. Ways of collecting data from view

Ans: Model object, form collection, request object, normal parameters in action method

39. Form Collection

The Form collection is used to retrieve the values of form elements from a form that uses the POST method.

Syntax:

Request.Form(element)[(index)|.Count]

40. Explain View Result and redirect result

ViewResult

View result is a basic view result. It returns basic results to view page. View result can return data to view page through which class is defined in the model. View page is a simple HTML page

RedirectResult

RedirectResult is an ActionResult that returns a Found (302), Moved Permanently (301), Temporary Redirect (307), or Permanent Redirect (308) response with a Location header to the supplied URL. It will redirect us to the provided URL, it doesn't matter if the URL is relative or absolute.

41. How to transfer data from controller to view?

Ans: view data, view bag and temp data

42. Can you delete a record form table using view?

We can, but if the view contains joins between multiple tables then we can only update one table in the view.

YES using DROP VIEW view_name syntax

43. What is ORM?

An ORM is an **application or system that support in the conversion of data within** a relational database management system (RDBMS) and the object model that is necessary for use within object-oriented programming. ORM products that supporting ASP.NET MVC : NHibernate, Entity Framework, Linq-to-SQL

44. Joins in EDM – navigational properties

45. What is the namespace of dbcontext?

System.Data.Entity

46. How to modify/update edmx if changes are made in the database?

How to create entity data model?

Update the .edmx file when the Database changes

1. In the Model Browser, right-click the .edmx file and select Update Model from Database.
2. After this we have to Expand the Tables, Views, and Stored Procedures nodes, and check the objects we want to add to the .edmx file.
3. Click the Add tab.
4. In add tab we can see the Nodes for tables, views, and stored procedures are displayed. If any objects have been added to the database we can expand the corresponding node to view the objects that are available to add to the conceptual model.

5. After this we have to Click the Refresh tab.
6. The Nodes for tables, views, and stored procedures that are included in the existing storage model are displayed. Any changes that have been made to these database objects will be reflected in the updated storage model.
7. After this we have to Click the Delete tab.
8. Nodes for tables, views, and stored procedures are displayed. If an object has been deleted from the database and was included in the previous storage model, we can expand the corresponding node. The objects in these nodes will be deleted from the updated model.
9. At last we have to Click Finish to update the .edmx file with the database changes.

Create Entity Model

1. Click on Project -> Add New Item.
2. Select Data from the left menu and then ADO.NET Entity Data Model.
3. Enter TestModel as the name and click OK.
4. This launches the Entity Data Model Wizard.
5. Select "Generate from database" and click Next.
6. Select the connection to the database and click Next.
7. Click the checkbox next to the 'Tables' to import all tables and click 'Finish'.
8. The new model is added to your project and opened for you to view in the Entity Framework Designer. An App.config file has also been added to your project with the connection details for the database.

47. Which approach other than db one?

There are three approaches to model your entities in Entity Framework: **Code First, Model First, and Database First**.

The **Code First approach** helps you to create the entities in your application by focusing on the domain requirements.
You can use the **Database First approach** if the database is already designed and is ready. In this approach, the Entity Data Model (EDM) is created from the underlying database.

In the **Model First approach** you can create the EDM first, then generate the database from it.

48. In the scenario to search which one do you choose linq or lambda expression?
49. Template for searching data by id?

50. Just after getting logged how can we immediately navigate to another page?

By clicking the URLs of our choice.
51. Mapping Model
52. How do you connect to any other database from ado.net?

. Click the **Connections** tab .
. Click **New connection** and choose **Database** from the menu.

The **New connection** window appears.

Choose the database type you want to connect to.Provide the connection properties for your database. Required fields are marked with an asterisk (*).

. Click **Add**.

53. Sql connection in ado.net
54. Where will u find route config?
55. Can we directly return json?
56. Return type in action method

In your action method, return Json(object) to return JSON to **your** page

57. Explain jsonresult

JsonResult is an ActionResult type in MVC. It helps to send the content in (JSON) format.

58. Can we return json through viewresult?
59. Hyperlink in mvc view?
Used for moving from one view to another. We can use html. ActionLink() inside the view code to achieve this.

60. Can error.cshtml will be one file or can be more?
61. FirstOrDefault and First
The difference btw these two is that First () will throw an exception if there is no result data for the supplied criteria where as FirstOrDefault () will return the default value (null) if there is no result data

62. If you want to set view as default view (i.e., it should get returned first when we execute our program) Where can we place that view?
We should see which action method controller it is returning first, put return View () in that method
(Or)
You need to set default controller and action method in route config file.

VIEW	PARTIAL VIEW
Contains layout page	Doesnot contain layout page

A view might have markup tags like body,HTML,Head,Title	It doesnot consist of any markup

63. Scaffolding templates

Create , details, list, delete , update, empty

BENEFIT OF USING MVC

SUPPORT OF MULTIPLE VIEWS
FASTER DEVELOPMENT PROCESS
LIGHTWEIGHT
MORE CONTROL

SCAFFOLDING- IS USED IN DEVELOPING MVC APPLICATIONS WHEN ANYONE WANTS TO RAPIDLY ENHANCE THE CODE THAT INTERMINGLES WITH THE APPLICATIONS DATA MODEL

MVC AUTHENTICATION-IN ORDER TO INCLUDE A LAYER OF SECURITY TO ACCESS THE USER FOR A SPECIFIC SERVICE.VERIFIES USERS IDENTITY CREDENTIALS SUCH AS USERNAME AND PASSWORD.

VIEW MODEL- PLAIN CLASS HAVING DIFFERENT PROPERTIES,IT IS USED FOR BINDING A VIEW THAT IS STRONGLY TYPED.

What is MVC?

MVC is a software architecture/application design model containing 3 interconnected portions.These 3 portions are:

Model(data associated with the application)

View(which is the user interface of MVC application

Controller(processes that are responsible for handling the I/P)

MVC model is used to develop modern applications with user interfaces.

Model- the data that will be used by the program.Commonly used examples of models in MVC are:

Database,a simple object holding data.

A view is a way of displaying objects within an application.This is the particular portion through which end users will communicate.

A controller is the third vertical which is responsible for updating both models and views.It accepts I/P from users as well as performs equivalent update.

Return types used by controller action method in MVC:

ViewResult

JSON result

Content Result

Redirect Result

Javascript Result

PartialView Result

EmptyResult

HTML FORM ELEMENTS

- **BEGINFORM()**
- **ENDFORM()**
- **TEXTAREA()**
- **TEXTBOX()**
- **CHECKBOX()**
- **RADIOBUTTON()**
- **LISTBOX()**
- **DROPDOWNLIST()**
- **HIDDEN()**
- **PASSWORD()**

MVC FILTERS

ASP.NET MVC Filter is a custom class where we can write custom logic to execute before or after an action method executes. Filters can be applied to an action method or controller.

Filter Type	Description	Built-in Filter	Interface
Authorization filters	Performs authentication and authorizes before executing an action method.	[Authorize], [RequireHttps]	IAuthorizationFilter
Action filters	Performs some operation before and after an action method executes.		IActionFilter
Result filters	Performs some operation before or after the execution of the view.	[OutputCache]	IResultFilter
Exception filters	Performs some operation if there is an unhandled exception thrown during the execution of the ASP.NET MVC pipeline.	[HandleError]	IExceptionFilter

PARTIAL VIEW- VIEW WHICH CAN BE RENDERED FROM ANOTHER VIEW WHICH IS CALLED PARENT VIEW.

HOW TO MAINTAIN SESSIONS IN MVC?
VIEW DATA,TEMP DATA,VIEW BAG

Some of The Most Important SQL Commands

- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index

SQL JOIN

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

- (INNER) JOIN: Returns records that have matching values in both tables
- LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
- RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
- FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table

The LEFT JOIN keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match.

The RIGHT JOIN keyword returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side, if there is no match.

A self join is a regular join, but the table is joined with itself.

SQL ALIASES
SQL aliases are used to give a table, or a column in a table, a temporary name.

The SQL BETWEEN Operator

The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.

SELECT *column_name(s)*
FROM *table_name*
WHERE *column_name* **BETWEEN** *value1* **AND** *value2;*

Six Default databases in SQL Server

SQL Server consists of six databases by default.

Master	**It contains system catalogs that keep information about disk space, file allocations, usage, system wide configuration settings, login accounts, the existence of other database, and the existence of other SQL Servers (for distributed operations).**
Model	**It is a simply a template database. Every time you create a new database, SQL Server makes a copy of model to form the basis of the new database.**
Tempdb	**Temporary database, tempdb, is a workspace. SQL Server's tempdb database is unique among all other databases because it is recreated not recovered every time SQL Server is started.**
Pubs	**This is a sample database used extensively by much of SQL Server documentation. It's available to everyone in the SQL Server community**
Northwind	**This is a sample database that was originally developed for the use of Microsoft Access.**
Msdb	**This database is used by the SQL Server Agent Service, which performs scheduled activities such as backups and replication tasks.**

CREATE DATABASE:CREATE DATABASE *databasename*;

DROP DATABASE statement is used to drop an existing SQL database:
DROP DATABASE *databasename*;

CREATE TABLE statement is used to create a new table in a database.

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
```

....
);

DROP TABLE statement is used to drop an existing table in a database
DROP TABLE *table_name*;

TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself
TRUNCATE TABLE *table_name*;

ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

ALTER TABLE *table_name*
ADD *column_name datatype*;

To delete a column in a table, use the following syntax
ALTER TABLE *table_name*
DROP COLUMN *column_name*;

To change the data type of a column in a table
ALTER TABLE *table_name*
ALTER COLUMN *column_name datatype*;

The SQL COUNT(), AVG() and SUM() Functions

The COUNT() function returns the number of rows that matches a specified criterion.

COUNT() Syntax

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

The AVG() function returns the average value of a numeric column.

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

The SUM() function returns the total sum of a numeric column.

SUM() Syntax

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

SQL Constraints

SQL constraints are used to specify rules for the data in a table.

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

> Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

The following constraints are commonly used in SQL:

- NOT NULL - Ensures that a column cannot have a NULL value
- UNIQUE - Ensures that all values in a column are different
- PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- FOREIGN KEY - Prevents actions that would destroy links between tables
- CHECK - Ensures that the values in a column satisfies a specific condition
- DEFAULT - Sets a default value for a column if no value is specified
- CREATE INDEX - Used to create and retrieve data from the database very quickly

```
UNIQUE---CREATE TABLE Persons (
    ID int NOT NULL UNIQUE,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int
);

PRIMARY KEY----CREATE TABLE Persons (
    ID int NOT NULL,
```

```
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    PRIMARY KEY (ID)
);
```

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.
The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

```
CREATE TABLE Orders (
    OrderID int NOT NULL,
    OrderNumber int NOT NULL,
    PersonID int,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)
);
```

To drop a FOREIGN KEY constraint, use the following SQL:

```
ALTER TABLE Orders
DROP FOREIGN KEY FK_PersonOrder;
```

SQL CHECK Constraint

The CHECK constraint is used to limit the value range that can be placed in a column.

If you define a CHECK constraint on a column it will allow only certain values for this column.

If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

```
ALTER TABLE Persons
ADD CHECK (Age>=18);
```

To drop a CHECK constraint, use the following SQL:

```
ALTER TABLE Persons
DROP CONSTRAINT CHK_PersonAge;
```

To create a DEFAULT constraint on the "City" column when the table is already created

```
ALTER TABLE Persons
ALTER City SET DEFAULT 'Sandnes';
```

The DROP INDEX statement is used to delete an index in a table.

```
DROP INDEX table_name.index_name;
```

SQL CREATE VIEW Statement

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

SQL Updating a View

```
CREATE OR REPLACE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

SQL Dropping a View

```
DROP VIEW view_name;
```

LIMIT	Specifies the number of records to return in the result set
OR	Includes rows where either condition is true
DROP CONSTRAINT	Deletes a UNIQUE, PRIMARY KEY, FOREIGN KEY, or CHECK constraint
DROP DATABASE	Deletes an existing SQL database
DROP DEFAULT	Deletes a DEFAULT constraint
DROP INDEX	Deletes an index in a table

DROP TABLE	Deletes an existing table in the database
DROP VIEW	Deletes a view
EXEC	Executes a stored procedure
DESC	Sorts the result set in descending order
CREATE PROCEDURE	Creates a stored procedure

SCENARIO:

The scenario mentions a leading software company, Acc Corp which has global presence.It offers insurance scheme to its employees, which includes insurance coverage for employees,their spouse,and children.As an additional benefit, the company decided to offer its employees a chance to nominate their parents/parents in law also for insurance coverage.
The employees can also opt to increase their insurance coverage for a additional amount deducted from their salary.An employee can nominate many relations as dependents but can have only a single coverage amount.
Acc corp wants to deploy MVC app to:

a. **login with their corporate ID**
b. **View dependent information**
c. **View insurance coverage amount and deductions**
d. **Add dependent information**
e. **Increase insurance coverage**

ACTION VARIABLES USED:

- **LOGIN**
- **VIEW DEPENDENT INFO**
- **VIEW COVERAGE INFO**
- **INCREASE COVERAGE**
- **ADD DEPENDENT INFO**