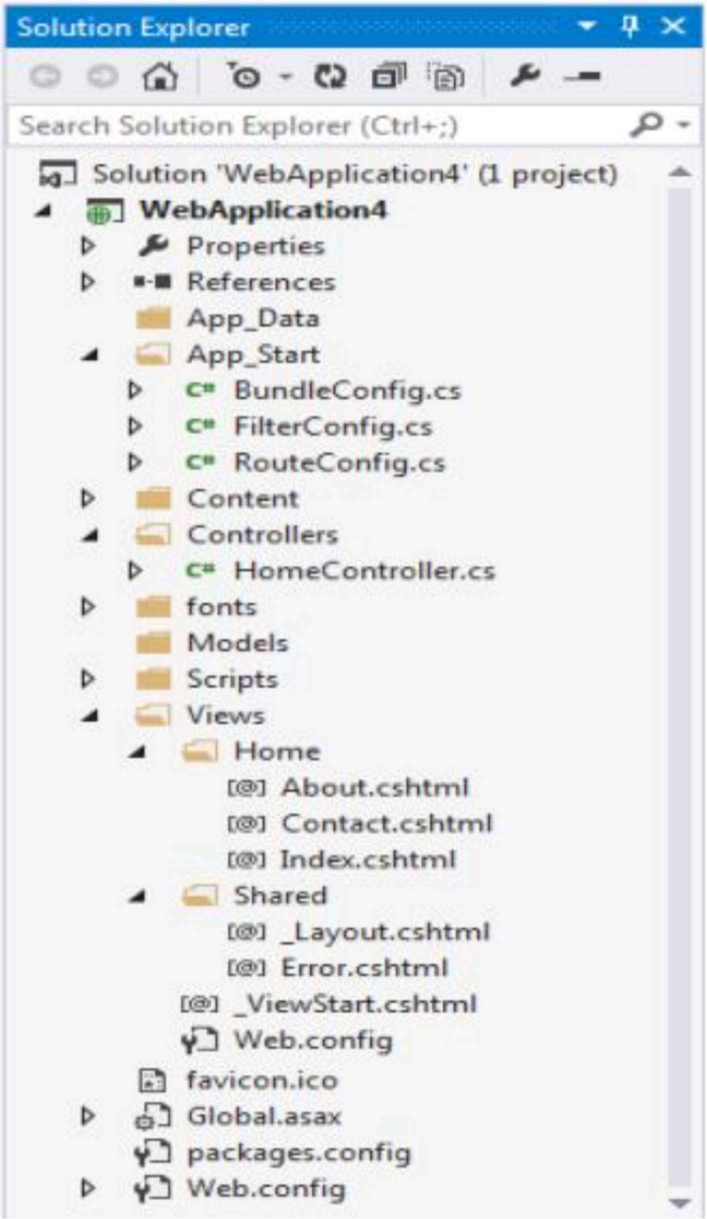

```
Solution 'WebApplication4' (1 project)
◢ WebApplication4
    ▷ Properties
    ▷ References
      App_Data
    ◢ App_Start
        ▷ BundleConfig.cs
        ▷ FilterConfig.cs
        ▷ RouteConfig.cs
    ▷ Content
    ◢ Controllers
        ▷ HomeController.cs
    ▷ fonts
      Models
    ▷ Scripts
    ◢ Views
        ◢ Home
              About.cshtml
              Contact.cshtml
              Index.cshtml
        ◢ Shared
              _Layout.cshtml
              Error.cshtml
          _ViewStart.cshtml
          Web.config
      favicon.ico
    ▷ Global.asax
      packages.config
    ▷ Web.config
```

MVC Framework Layout:

App Data:

Used to store application data such as portable databases (e.g., SQL Server Compact Edition) or other data files (XML, JSON, etc.). Typically, it’s best to use a full-fledged database like SQL Server.

App Start:

Contains the configuration and initialization of various features of the application.

BundleConfig.cs: Handles the configuration of bundling and minification of JavaScript and CSS files, reducing the number of requests for these resources.

FilterConfig.cs: Registers global filters like authorization or exception handling filters.

RouteConfig.cs: Configures routes to map URLs to controllers and actions in the application.

Additional configuration files like WebApiConfig.cs may be added when using other technologies (e.g., Web API).

Controllers:

Stores the controller classes that manage the logic of your application, handle user input, and return views or data.

Models:

Contains the business models that represent the data in your application.

Best practice: Use a separate project for models to promote separation of concerns. For simplicity in demonstrations, models can be stored in this folder.

Views:

Contains HTML Views with Razor syntax (.cshtml files) for rendering HTML pages.

Views are organized by controller. Each controller has its own folder under Views/ControllerName.

Shared Folder: Holds views that are shared across different controllers (e.g., layout pages, error pages).

Scripts:

Stores JavaScript files for client-side functionality. Can include libraries, plugins, or custom scripts.

Global.asax

A special file for application-wide initialization, such as configuring routes, bundles, and global filters. It's where the RouteConfig, BundleConfig, and FilterConfig are called when the application starts.

Web.Config

Contains configuration settings for the application:

<appSettings>: Stores general settings for your application.

<connectionStrings>: Stores database connection strings and other external resources.

Additional Considerations

WebApiConfig.cs: When adding Web API to your MVC project, this file is added to configure Web API routes.

Error Pages: Use shared views for error handling across the application, ensuring users see consistent error messages.