This sample application provides a web service for users to register, login and search for cars based on common criteria such as the make, manufactured year, color and more. The application itself is built using the ASP.NET MVC framework.

It has the following features:

|  |  |
| --- | --- |
| **User Management** | |
| Registration | In order to search for vehicles, the users will need to register with the site by providing an email and password. The service validates the email and password for strength and stores in the database. Passwords are stored as a hash in the database and can be reset by the users. |
| Authentication | In order to search for cars, the users will have to login to the portal using their email ID and password already registered with the site. |
| **Website feature highlights** | |
| List all available vehicles | The website default view lists all inventory stored in the database. |
| Validations | When adding new vehicles, validation is performed to maintain data integrity. All vehicles require the vehicle Make – for e.g., Toyota, Chevrolet, etc.  The manufactured year is validated for the date format.  Only certain vehicle types are allowed. |
| View vehicles | Images of the vehicles is mandatory and displayed in the default list view of all the vehicles. |
| Pagination | Given the potential for a large list of vehicles, the website provides a paginated view. The number of vehicles per page is fixed in the current implementation, but could be enhanced to allow the user to configure the number of entries per page. |
| Search and sort capabilities | Given the potential large list of vehicles, the current implementation allows to search for vehicles by Make.  The list can be sorted by specific fields such as Make, Year of manufacture, Image of the vehicle. |
| Inventory management | Vehicles added to the inventory can be edited or deleted |
|  | When adding or editing inventory, the web service provides the following validation:   1. Vehicle make is limited to a pre-defined set (for e.g., Toyota, Chevrolet, Honda, Ford etc.) 2. Date of manufacture is validated for basic format 3. Images are mandatory for all vehicles |
| Database implementation details for vehicles | The webservice uses a SqlExpress LocalDB for this implementation. The database consists of the following tables:   1. Cars – contains the following properties – Make, Color, Year, Owner Last Name, Image File, Description 2. Car Types – Type Name   The Cars table contains a n..1 relationship to Car Types. |

|  |  |
| --- | --- |
| Authentication | In order to search for cars, the users will have to login to the portal. |
| Registration would be nice but not required. | Done |
| Simple Object Call | Created a car list. |
| Required name required type of thing. | Implemented validations on the data fields |
| With Images | Yes. Image is being saved to the database. |
| Relationship | Has parent child data model relationship |
| Paginated List want to see a paginated list with image and name. | Yes. It has paginated list. You can go previous, next etc. |
| Search by name | Has a separate search page as well as in the application page |
| edit and create – delete from list. | Yes. It does the CRUD operations |
| edit/create page should have basic validation - required, max length. image is required. | It has validations on number of data fields. |
|  | Also added sorting by various data fields |
|  |  |