

ACKNOWLEDGEMENT

It is with immense pleasure and satisfaction that we present this project report on **DR Car Parking** let us first express our sincere to *MISS NISHITA MAM* for the kind co-operation extended to us by giving opportunity to work on this project in their esteemed organization.

We are heartily grateful to *MISS SUMAN MAM* for her kind response and co-operation..

INDEX

NO	CONTENTS	PAGE NO.
1	Introduction	3
2	Parts of Visual Basic	5
3	Introduction to project	6
4	Description of controls	7
5	Screenshots	13

INTRODUCTION

⇒ **About Visual Basic 6.0**

Visual Basic developed from the old Quick Basic language that was available under the DOS operating system. Quick Basic is the same language as BASIC; it is just Microsoft's product name for its version of BASIC. Visual Basic started out as Brainchild of Alan Cooper. Cooper developed Visual Basic and then sold the product to Microsoft. Microsoft took the undeveloped product code-named it "Thunder,; the proceeded to create a programming language that would soon become one of the premier development environments in the windows environment.

⇒ **Features of Visual Basic**

- Visual Basic is a superset of the Visual Basic for Application (VBA) programming language, which is included with most of the office products.
- Include a GUI development environment for development Windows applications.
- Provide the ability to develop and test application using an interpretive run function.
- Allows for the creation of p-code and native code EXE files. P-code is a tokenized form of your source code that will be broken down at running into machine code.
- P-code can be stored more efficiently and executed faster than your source code, which is why Visual Basic will create this intermediary form.
- Object-based development is possible using class modules.
- Rapid application development (RAD).
- Allows for the creation of COM components such as ActiveX controls, DLLs and EXEs.

- Has many internet development possibilities, including following :
 1. ActiveX documents
 2. DHTML applications that help you test browser application.
 3. IIS applications that help you build server-based application.
 4. Web browser control.
 5. FTP/HTTP support through a custom control.
 6. Winsock control.
- Has an excellent integrated help facility and books online. Including good debugging facilities. It has many wizards that help automate repetitive tasks.
- Can be extended easily through the use of Windows API calls, hundreds of third party controls and DLLs, and integration with other Windows.
- Uses many database access methods to get at different types of data.

⇒ Visual Basic Advantages

- Visual Basic has many advantages over other development languages.
- Here's the list of some advantages :
- Has a shorter learning curve and development time than C/C++, Delphi and even Power Builder.
- Removes the complexities of the windows API from the programmer.
- Allows for rapid application development.
- Excellent for business application.
- Used by most of the office site tools as their macro language, with the Help of follow. Other companies as well are starting to support VBA in their product, such as AutoCad, Corel Draw, SAP, and many others.
- Allows you to create ActiveX controls.

Parts of Visual Basic

Forms

Forms are familiar to all visual basic programmers, of course they are the templates you base windows on.

Modules

Modules are collection of code and data that function something like objects in object-oriented programming (OOP), characteristics like inheritance polymorphism, and so on.

Global items

Global items are accessible to all modules and forms in a project, and you declare them with the public keyword. Microsoft recommends to keep the number of global items to an absolute minimum, in fact, suggests their use only when you need to communicate between forms.

INTRODUCTION TO PROJECT

- The project DR Car Parking is very useful in many places.
- It will store parking information of car.
- It stores the status, number and name of car in database.
- By using this project you can add new car into park, search for a car, and remove car.
- We can also clear whole database if we want.
- For storing a new car we've to enter car number & car name.
- This system is used to search parked car.

DESCRIPTION OF CONTROLS

- **FORM**

Form Window

Allows you to create the windows, dialog boxes, and controls in your [application](#). You draw and view [controls](#) on a [form](#).

While you are designing a form:

- Each form window has a Maximize, Minimize, and Close button.
- You can create either fixed or movable forms. The form you design will have the same features at [design time](#) and at [run time](#) unless you specify otherwise in the form's properties.
- Use the buttons in the Toolbox to draw controls on the form.
- Use the Form Layout Window command from the View menu to preview the layout of your form on a screen.

Caption Property

- **Form** — determines the text displayed in the **Form** or **MDIForm** object's [title bar](#). When the form is minimized, this text is displayed below the form's [icon](#).
- **Control** — determines the text displayed in or next to a control.
- **MenuLine** object — determines the text displayed for a **Menu** control or an object in the **MenuItems** collection.

For a **Menu** control, **Caption** is normally read/write at [run time](#). But **Caption** is read-only for menus that are exposed or supplied by Visual Basic to add-ins, such as the **MenuLine** object.

Syntax

object.**Caption** [= *string*]

The **Caption** property syntax has these parts:

Part	Description
------	-------------

<i>Object</i>	An object expression that evaluates to an object in the Applies To list. If <i>object</i> is omitted, the form associated with the active form module is assumed to be <i>object</i> .
<i>String</i>	A string expression that evaluates to the text displayed as the caption.

Remarks

When you create a new object, its default caption is the default **Name** property setting. This default caption includes the object name and an integer, such as Command1 or Form1. For a more descriptive label, set the **Caption** property.

You can use the **Caption** property to assign an [access key](#) to a control. In the caption, include an ampersand (&) immediately preceding the character you want to designate as an access key. The character is underlined. Press the ALT key plus the underlined character to move the [focus](#) to that control. To include an ampersand in a caption without creating an access key, include two ampersands (&&). A single ampersand is displayed in the caption and no characters are underlined.

A **Label** control's caption size is unlimited. For forms and all other controls that have captions, the limit is 255 characters.

To display the caption for a form, set the **BorderStyle** property to either Fixed Single (1 or **vbFixedSingle**), Sizable (2 or **vbSizable**), or Fixed Dialog (3 or **vbFixedDialog**). A caption too long for the form's title bar is clipped. When an [MDI child](#) form is maximized within an **MDIForm** object, the child form's caption is included in the parent form's caption.

Tip For a label, set the **AutoSize** property to **True** to automatically resize the control to fit its caption.

MDIChild Property

Returns or sets a value indicating whether a form is displayed as an [MDI child](#) form inside an [MDI form](#). Read only at [run time](#).

Syntax

object.**MDIChild**

The *object* placeholder represents an [object expression](#) that evaluates to an object in the Applies To list.

Settings

The **MDIChild** property settings are:

Setting	Description
True	The form is an MDI child form and is displayed inside the parent MDI form.
False	(Default) The form isn't an MDI child form.

Remarks

Use this property when creating a multiple-document interface (MDI) application. At [run time](#), forms with this property set to **True** are displayed inside an MDI form. An MDI child form can be maximized, minimized, and moved, all inside the parent MDI form.

When working with MDI child forms, keep the following in mind:

- At run time, when an MDI child form is maximized, its caption is combined with that of the parent MDI form.
- At design time, an MDI child form is displayed like any other form because the form is displayed inside the parent form only at run time. An MDI child form's icon in the Project window is different from icons for other kinds of forms.
- MDI child forms can't be [modal](#).
- The initial size and placement of MDI child forms are controlled by the Microsoft Windows operating environment unless you specifically set them in the Load event procedure.
- If an MDI child form is referenced before the parent is loaded, the parent MDI form is automatically loaded. However, if the parent MDI form is referenced before loading an MDI child form, the child form isn't loaded.

Note All MDI child forms have sizable borders, a Control-menu box, and Minimize and Maximize buttons, regardless of the settings of the **BorderStyle**, **ControlBox**, **MinButton**, and **MaxButton** properties.

Any reference to an **MDIForm** object, including reading or setting properties, causes the form to load and become visible.

Picture Property

Returns or sets a graphic to be displayed in a control. For the **OLE** container control, not available at [design time](#) and read-only at [run time](#).

Syntax

```
object.Picture [= picture]
```

The **Picture** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>picture</i>	A string expression specifying a file containing a graphic, as described in Settings.

Settings

The settings for *picture* are:

Setting	Description
(None)	(Default) No picture.
(Bitmap, icon, metafile, GIF, JPEG)	Specifies a graphic. You can load the graphic from the Properties window at design time. At run time, you can also set this property using the LoadPicture function on a bitmap , icon , or metafile .

Remarks

At design time, you can transfer a graphic with the Clipboard using the Copy, Cut, and Paste commands on the Edit menu. At run time, you can use Clipboard methods such as **GetData**, **SetData**, and **GetFormat** with the nontext Clipboard [constants](#) **vbCFBitmap**,

vbCFMetafile, and **vbCFDIB**, which are listed in the Visual Basic (VB) [object library](#) in the [Object Browser](#).

When setting the **Picture** property at design time, the graphic is saved and loaded with the form. If you create an [executable file](#), the file contains the image. When you load a graphic at run time, the graphic isn't saved with the [application](#). Use the **SavePicture** statement to save a graphic from a form or picture box into a file.

Note At run time, the **Picture** property can be set to any other object's **DragIcon**, **Icon**, **Image**, or **Picture** property, or you can assign it the graphic returned by the **LoadPicture** function.

Note The Unisys Corporation has a patent that it alleges covers certain aspects of GIF-LZW compression technology, which the PictureBox and Image controls use. Microsoft Corporation obtained a license to the Unisys LZW patents in September, 1996. Microsoft's license does not, however, extend to software developers or third parties who use any Microsoft toolkit, language development, or operating system products to provide GIF read/write and/or any other LZW capabilities in their own products (for example, by way of DLLs and APIs).

Text Box properties :

Text Property

- **ComboBox** control (**Style** property set to 0 [Dropdown Combo] or to 1 [Simple Combo]) and **TextBox** control — returns or sets the text contained in the edit area.
- **ComboBox** control (**Style** property set to 2 [Dropdown List]) and **ListBox** control — returns the selected item in the [list box](#); the value returned is always equivalent to the value returned by the expression `List(ListIndex)`. Read-only at [design time](#); read-only at [run time](#).

Syntax

```
object.Text [= string]
```

The **Text** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>string</i>	A string expression specifying text.

Remarks

At design time only, the defaults for the **Text** property are:

- **ComboBox** and **TextBox** controls — the control's **Name** property.
- **ListBox** control — a zero-length string ("").

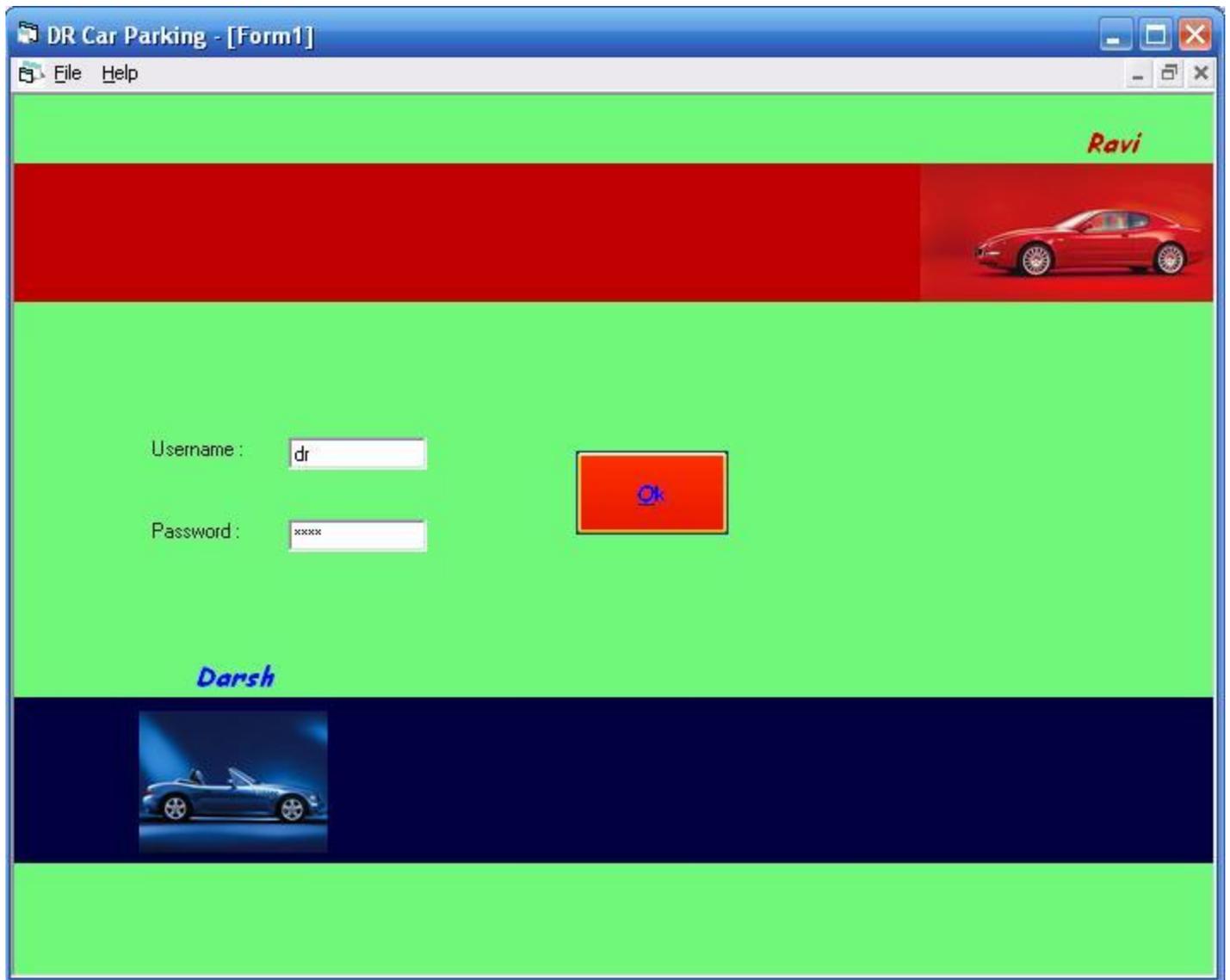
For a **ComboBox** with the **Style** property set to 0 (Dropdown Combo) or to 1 (Simple Combo) or for a **TextBox**, this property is useful for reading the actual string contained in the edit area of the control. For a **ComboBox** or **ListBox** control with the **Style** property set to 2 (Dropdown List), you can use the **Text** property to determine the currently selected item.

The **Text** setting for a **TextBox** control is limited to 2048 characters unless the **MultiLine** property is **True**, in which case the limit is about 32K.

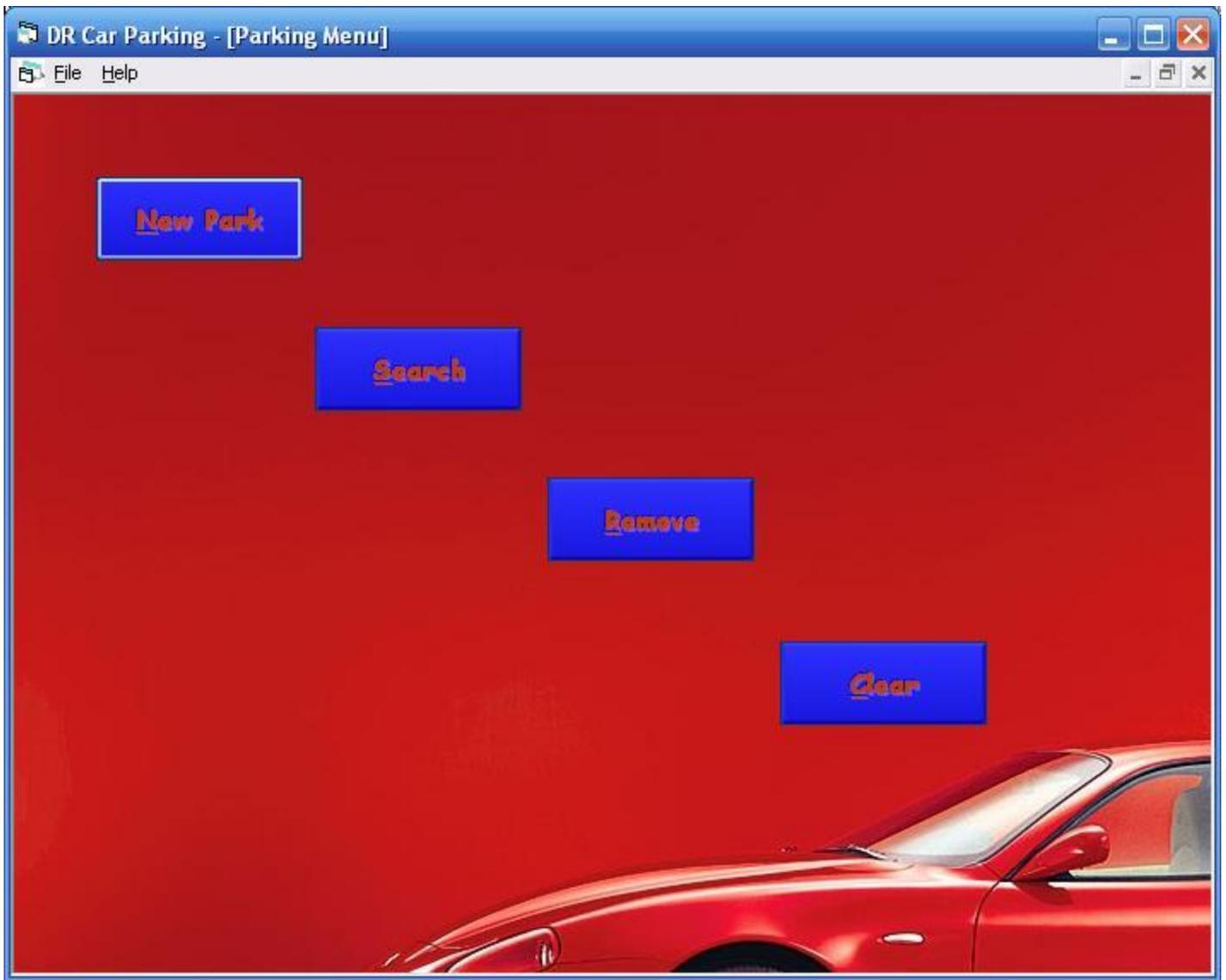
Screen shots of project :



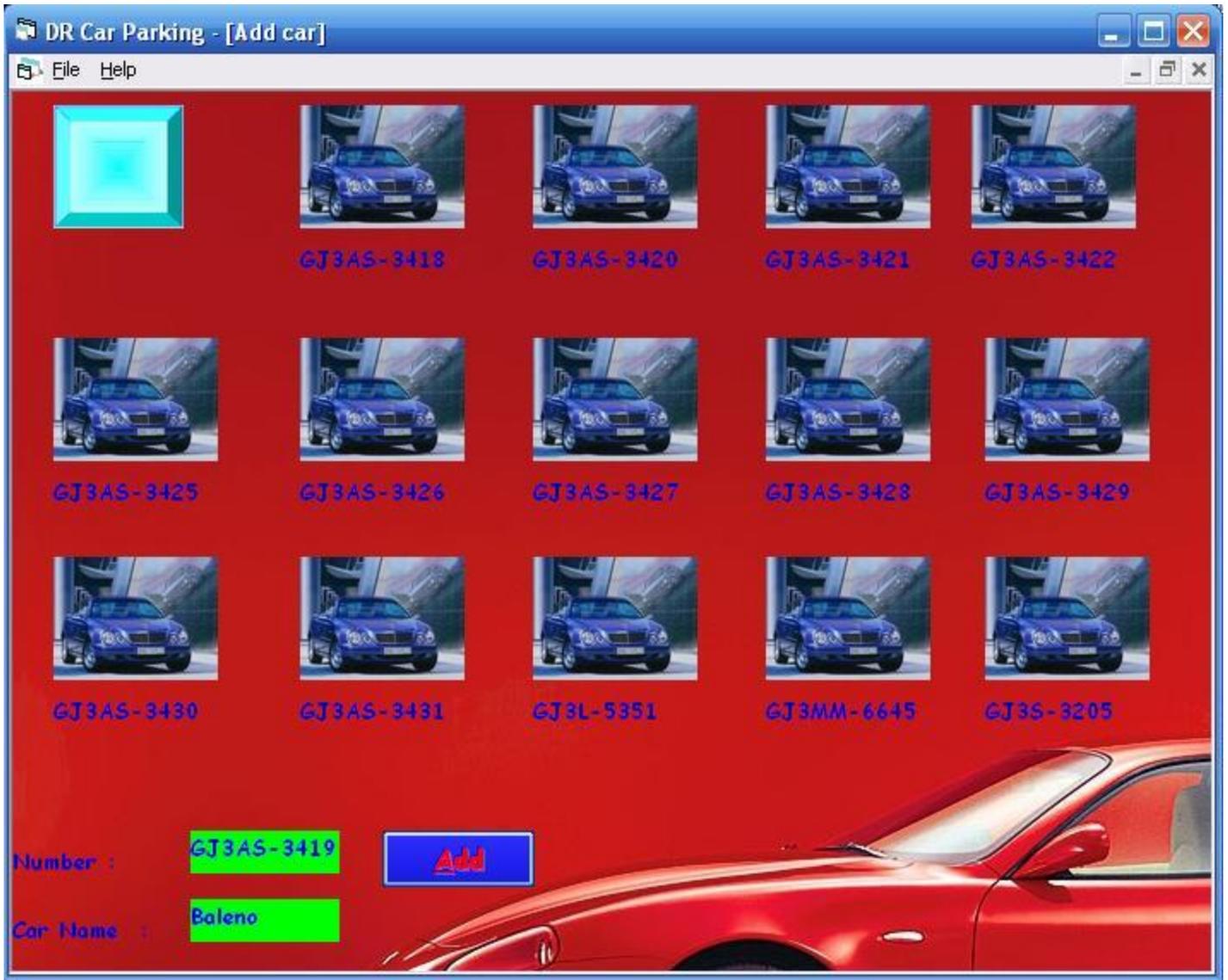
MDI form



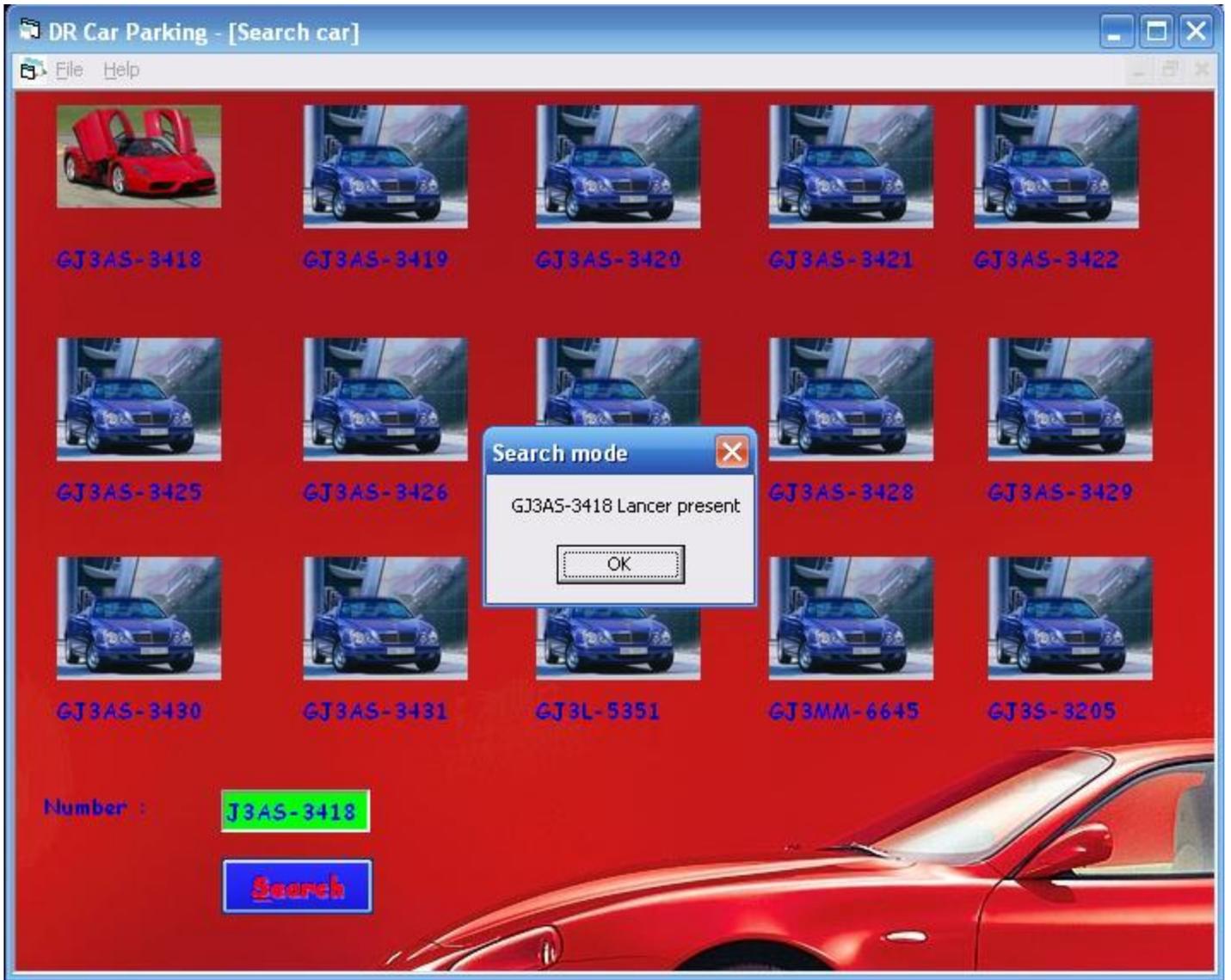
Login Screen



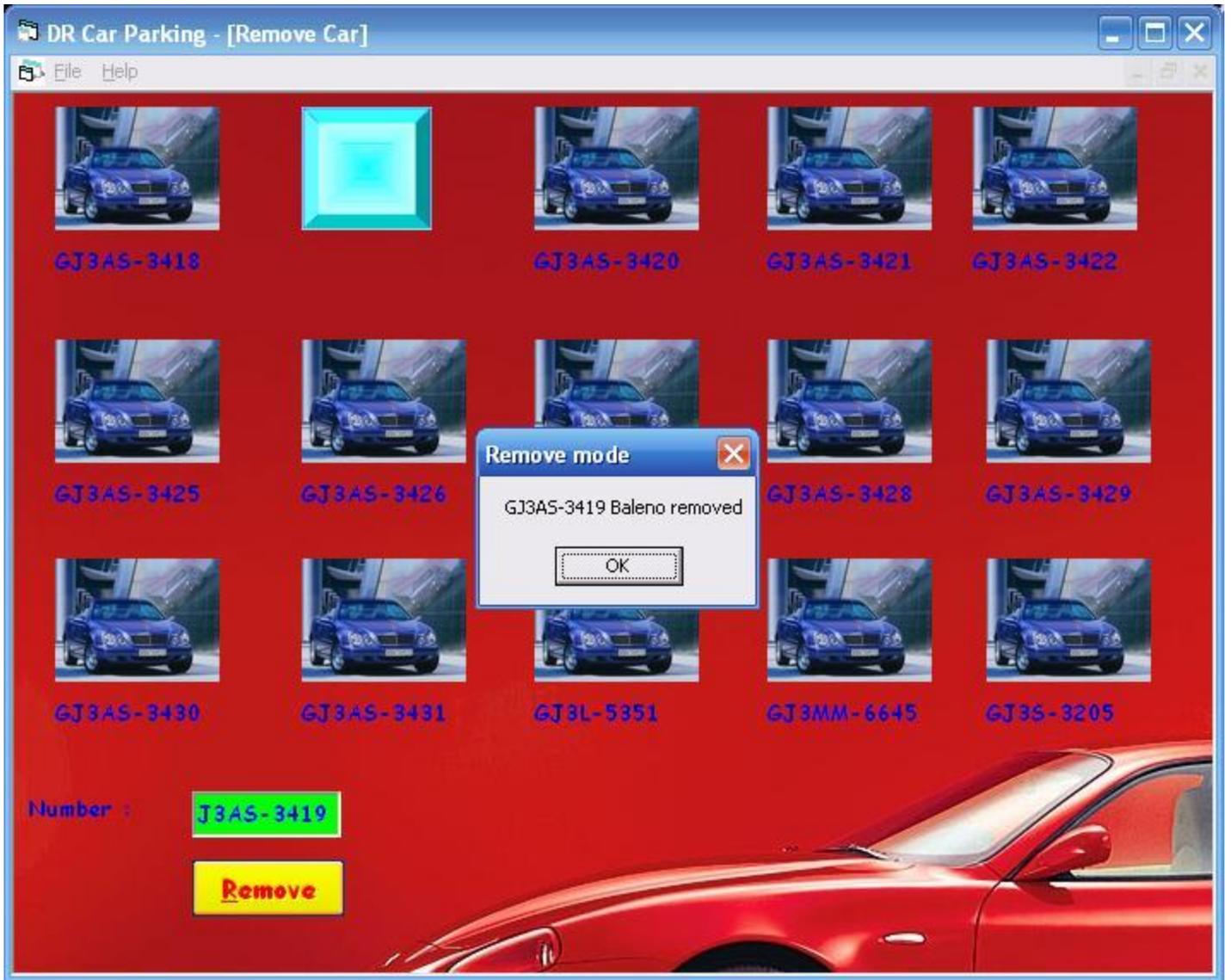
Parking Menu



Add Car



Search Car



Remove Car



Clear Database