**Real Estate Predictor**

**Business Case**

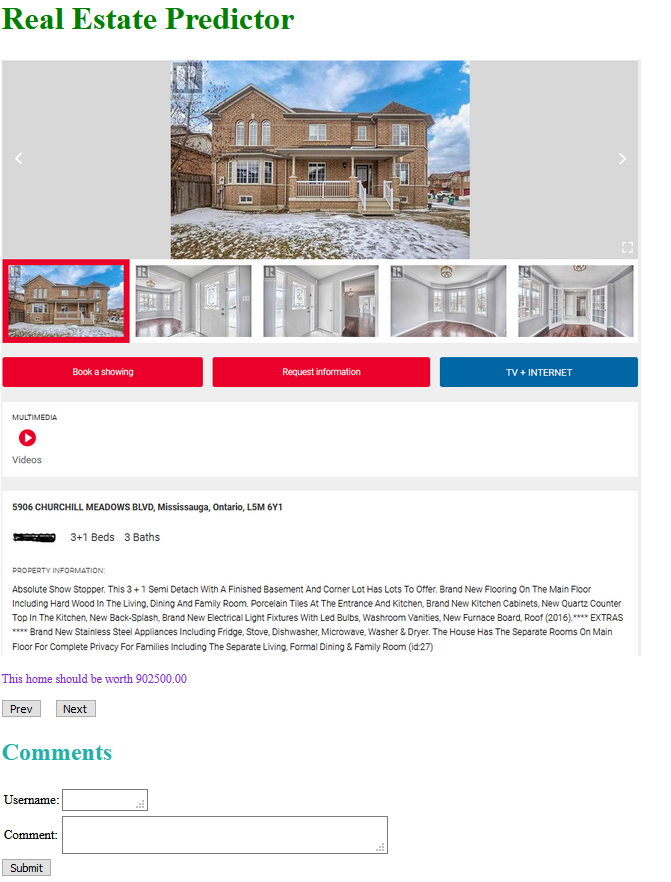
Customers in the real estate market like to have an idea of how much a house might cost. They like to discuss these ideas with each other. Much clarification is achieved this way, especially if an agent is part of the discussion.

**Requirements**

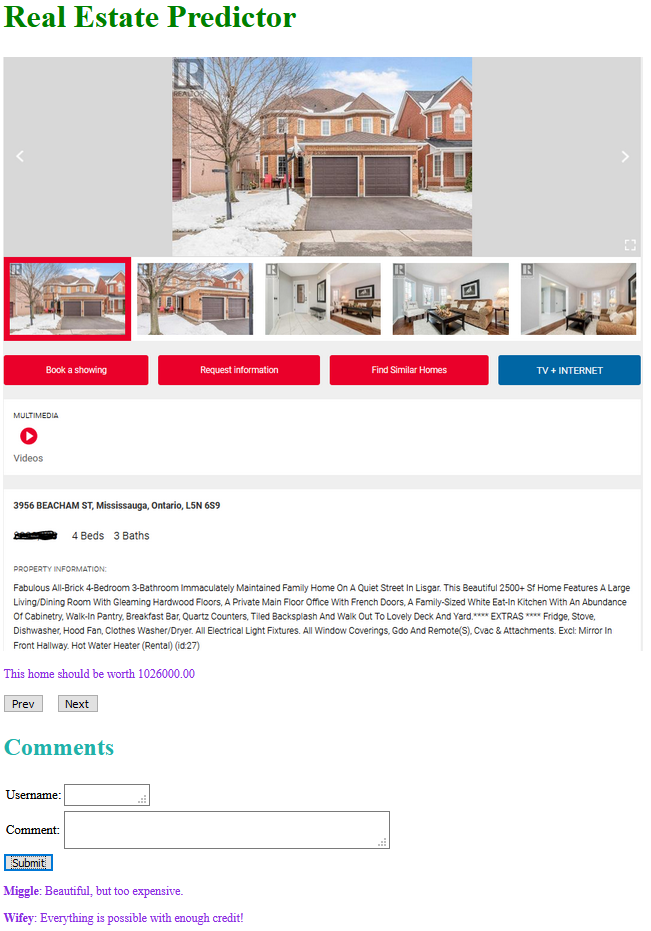
|  |  |  |
| --- | --- | --- |
| Requirement  Identifier | Description | Priority  (High, med, low) |
| RE-01 | The webpage must show an image of the house with its estimated value. | H |
| RE-02 | The webpage must have a PREV button to go to the previous image. | H |
| RE-03 | The webpage must have a NEXT button to go to the next image. | H |
| RE-04 | If we are at the first image, PREV will wrap around and take us to the last image. | H |
| RE-05 | If we are at the last image, NEXT will wrap around and take us to the first image. | H |
| RE-06 | Each webpage will have a comments section consisting of a username text area, a comment text area, and a submit button. | H |
| RE-07 | All submitted comments will be displayed below the submit button. | H |
| RE-08 | All submitted comments will display as username first and comment after. | H |
| RE-09 | All submitted comments will be permanent, they will not be deleted if we go to the next image. | H |
| RE-10 | Each webpage should be able to accommodate at least 1000 comments. | H |
| RE-11 | The username and comment text areas should clear after a submit. | H |
| RE-12 | The comments should be scrollable. | M |
| RE-13 | The comments should be searchable. | M |
| RE-14 | In any discussion, the real estate agent should be identifiable as such. | M |
| RE-15 | The comments could have a like button | L |
| RE-16 | The comments could keep track of the number of likes for each comment | L |
| RE-17 | The comments could have spell check. | L |

**Use-Case Scenarios**

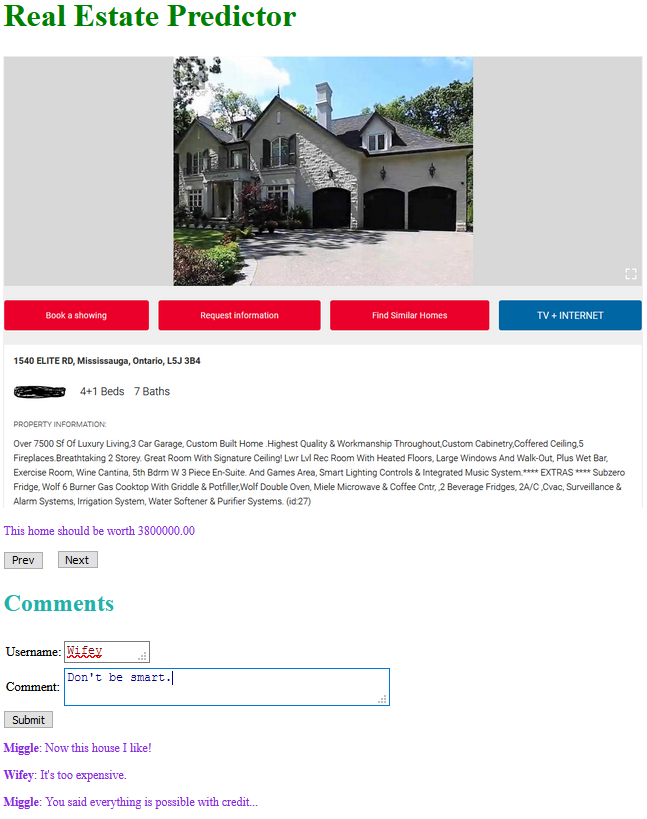
The webpage displays five houses one after another, using **Prev** and **Next** buttons. Each page will display a house with a description, an estimate of what the house is worth, and a comments section. The webpage for the first house is as follows:



As users scroll from one home to another they can add comments. Here is the second house with comments:



Here is the third house with comments. Note one comment has not yet been submitted:



If you go back to the second house, the comments should still be there.

**Design**

The HTML file consists of a header, an image of the house, an estimate of what the house is worth, **Prev** and **Next** buttons, and a comments section.

The JavaScript file has five variables:

* **homes[]** is the array of homes
* **index** is the index into the homes array. It indicates which home is currently on display.
* **numHomes** is a constant indicating how many homes are in the homes array.
* **images** is an array depicting the location of all the image files, assuming they are all located in an images subdirectory.
* **taxes** is the property tax for each home.

The function **CreateHomes()** creates all homes in the homes array as well as adds an event listener for the button Submit.

The **Homes** constructor accepts two parameters: the image of the home and the property tax.

The **Homes** constructor consists of the following variables:

* **desc** – the image showing and describing the house
* **tax** – the property tax for the house
* **value** – the estimated value of the house (calls **GetValue()**)
* **blogs[]** – an array of blogs (or comments)
* **numBlogs** – the number of blogs (or comments)

The **Homes** constructor will have one prototype function: **GetValue()**

**GetValue()** calculates the estimated value of the home by multiplying the property tax by 190.00 and rounding the result to two decimal places.

When a user wishes to add a comment, the user will enter a username, a comment, then click **Submit**.

The event listener for **Submit** will extract the username and the comment (bolding the username) and add it to the blogs[] array. It will then increment numBlogs. It will then clear the username and comments in the Comments section.

The function **Prev** will go to the previous house (decrements **index**). If we are at the first house, it will cycle to the fifth. The function **Next** will go to the next house (increments **index**). If we are at the fifth house, it will cycle to the first. Both **Prev** and **Next** will clear the username and comments in the Comments section.