Jan To Tong, Justin Nguyen, Rabin Gurung

Team 8

CS 157A Summer 2020

Functional Requirements

<u>How User will access the system:</u>

There will be two types of users: admins and consumers. Consumers would be able to access the system for input and output commands. Consumers are the targeted audience and the application will focus on giving the consumers the best experience by giving the consumers a list of recommended cars based on the user's input. Consumers will be able to browse cars in the database, request a list based on their input, view all the cars in the database, modify their input and list, and share their list with other users. Consumers will NOT be able to add or remove cars in the database. Admins would be able to do eve-rything the consumers could, add or remove cars from the database, and change the status of users. The users can access the application via a search engine to a website with an URL through the internet.

Functions:

Request list

- The user shall be able to search the database for cars by typing one or more characteristics of the car in a search box, such as the price floor/ceiling, make, min/max year in a search box.
- The system shall provide a car or cars that satisfy the search parameters. In the case that no car matches the search criteria, a message of “No car found” will appear.

Browse car

- The user shall be able to browse the database for cars. Also, the system shall allow the user to search for cars under specific criteria under a menu.
- The user shall be able to see a car’s characteristics by searching.

Display car information

- The user shall be able to display a car’s characteristic by searching for a car in the database

Add or Remove cars from the list

- The user shall be able to remove or add cars to their list by typing the car’s name in a search box.
- A user can only edit his or her own search box.

Share list

- A registered user shall be able to share his list with another registered user by filling in another user’s ID in a search box
- Every registered user shall have a unique ID upon registration

Create a list

- Users shall be able to create an empty list by clicking on a create list option.
- Users shall be able to add more than one car to their wish list.
- Users shall be able to add to or delete from their friend list.

Clone list

- A user shall be able to clone his own list or another registered user's list. Upon cloning, a new list would be created under the user who cloned the list.

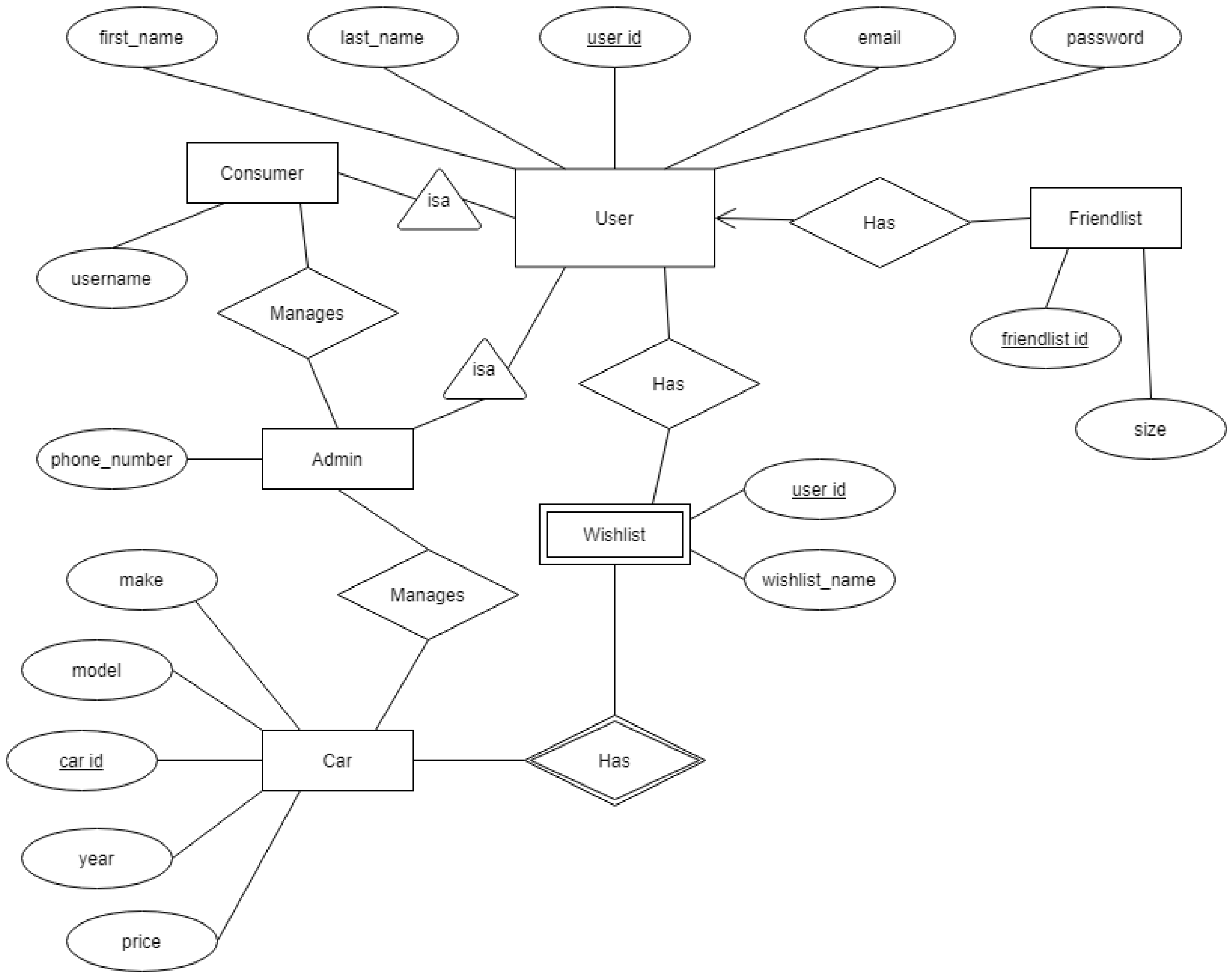
Creating user

- There are two types of users: Admin and consumer. Both types of accounts can be created. An admin will have more privileges than a consumer, such as adding/removing cars from the database and editing car characteristics.
- Admin however must be created through the database itself

Modifying database

- An admin, but not a consumer, shall be able to add or remove cars from the database.
- An admin can also edit the characteristics of a car by inputting new information on the website.
- An admin will also be able to edit a consumer's username in case there is some issues with it.

Project Design Section:



Database Schema

Database Name: Vehicle Selector

Database: Entity Set

1. User(<u>user_id</u>, first_name, last_name, email, password)
2. Admin(<u>user_id</u>, phone_number)
3. Consumer(<u>user_id</u>, username)

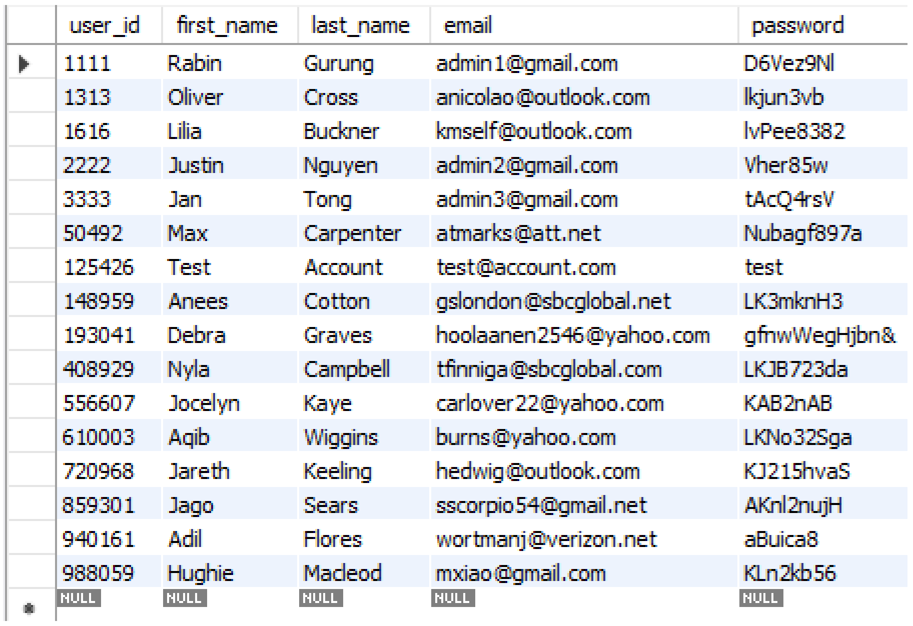
4. Wishlist(wishlist_name, size)
5. Friendlist(friendlist_id , size)
6. Car(car_id, make, model, year, price)

Database: Relationship

1. UserHasWishlist(user_id, wishlist_name)
2. AdminUpdatesCar(user_id, car_id)
3. AdminUpdatesConsumer(user_id, username)
4. UserHasFriendlist(friendlist_id , user_id)

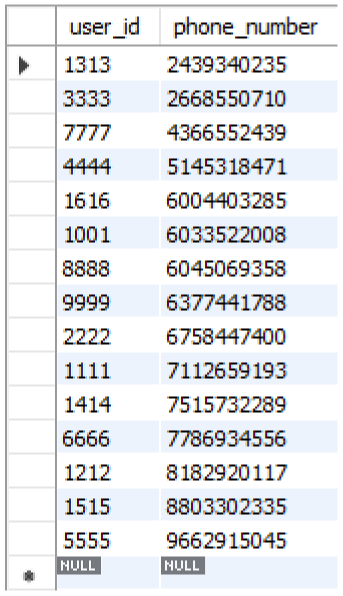
TABLES

User



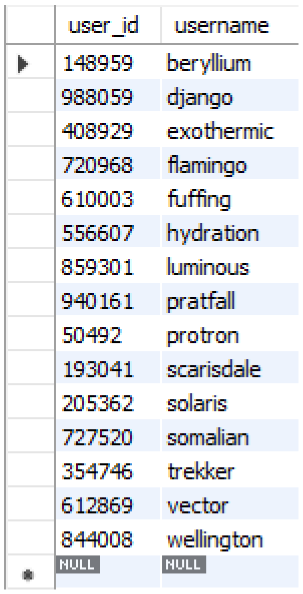
user_id	first_name	last_name	email	password
1111	Rabin	Gurung	admin1@gmail.com	D6Vez9Nl
1313	Oliver	Cross	anicolao@outlook.com	lkjun3vb
1616	Lilia	Buckner	kmself@outlook.com	lvPee8382
2222	Justin	Nguyen	admin2@gmail.com	Vher85w
3333	Jan	Tong	admin3@gmail.com	tAcQ4rsV
50492	Max	Carpenter	atmarks@att.net	Nubagf897a
125426	Test	Account	test@account.com	test
148959	Anees	Cotton	gslondon@sbcglobal.net	LK3mknH3
193041	Debra	Graves	hoolaanen2546@yahoo.com	gfnwWegHjbn&
408929	Nyla	Campbell	tfinniga@sbcglobal.com	LKJB723da
556607	Jocelyn	Kaye	carlover22@yahoo.com	KAB2nAB
610003	Aqib	Wiggins	burns@yahoo.com	LKNo32Sga
720968	Jareth	Keeling	hedwig@outlook.com	KJ215hvaS
859301	Jago	Sears	sscorpio54@gmail.net	AKnl2nujH
940161	Adil	Flores	wortmanj@verizon.net	aBuica8
988059	Hughie	Macleod	mxiao@gmail.com	KLn2kb56
NULL	NULL	NULL	NULL	NULL

Admin



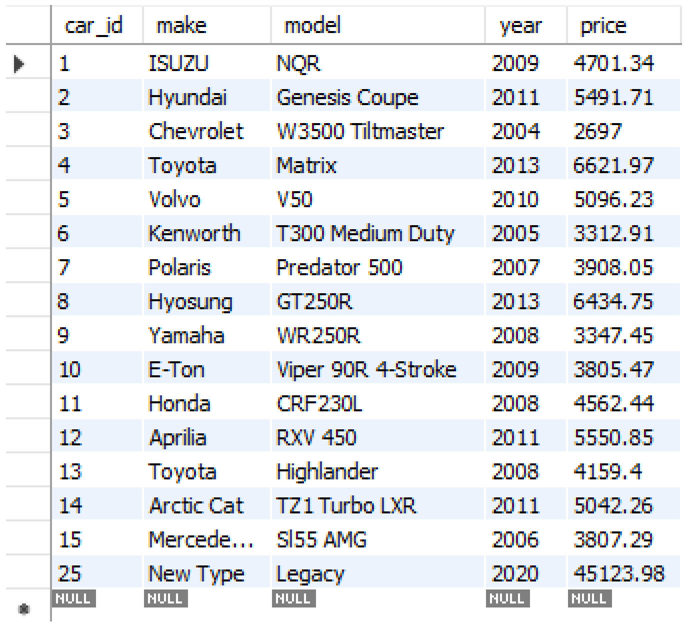
user_id	phone_number
1313	2439340235
3333	2668550710
7777	4366552439
4444	5145318471
1616	6004403285
1001	6033522008
8888	6045069358
9999	6377441788
2222	6758447400
1111	7112659193
1414	7515732289
6666	7786934556
1212	8182920117
1515	8803302335
5555	9662915045
NULL	NULL

Consumer



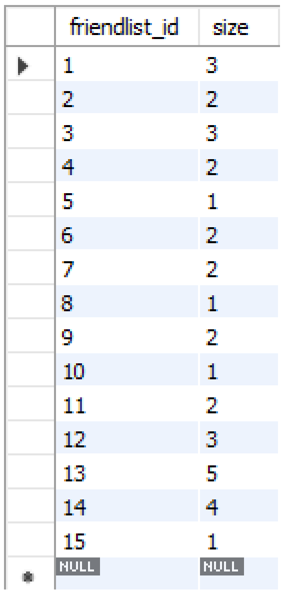
user_id	username
148959	beryllium
988059	django
408929	exothermic
720968	flamingo
610003	fuffing
556607	hydration
859301	luminous
940161	pratfall
50492	protron
193041	scarisdale
205362	solaris
727520	somalian
354746	trekker
612869	vector
844008	wellington
NULL	NULL

Car



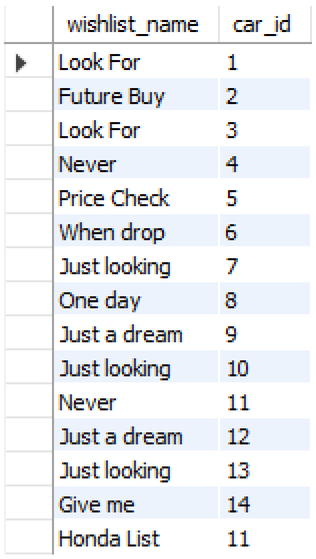
car_id	make	model	year	price
1	ISUZU	NQR	2009	4701.34
2	Hyundai	Genesis Coupe	2011	5491.71
3	Chevrolet	W3500 Tiltmaster	2004	2697
4	Toyota	Matrix	2013	6621.97
5	Volvo	V50	2010	5096.23
6	Kenworth	T300 Medium Duty	2005	3312.91
7	Polaris	Predator 500	2007	3908.05
8	Hyosung	GT250R	2013	6434.75
9	Yamaha	WR250R	2008	3347.45
10	E-Ton	Viper 90R 4-Stroke	2009	3805.47
11	Honda	CRF230L	2008	4562.44
12	Aprilia	RXV 450	2011	5550.85
13	Toyota	Highlander	2008	4159.4
14	Arctic Cat	TZ1 Turbo LXR	2011	5042.26
15	Mercede...	Sl55 AMG	2006	3807.29
25	New Type	Legacy	2020	45123.98
NULL	NULL	NULL	NULL	NULL

Friendlist



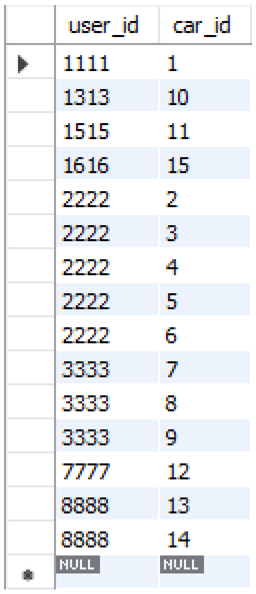
friendlist_id	size
1	3
2	2
3	3
4	2
5	1
6	2
7	2
8	1
9	2
10	1
11	2
12	3
13	5
14	4
15	1
NULL	NULL

Wishlist



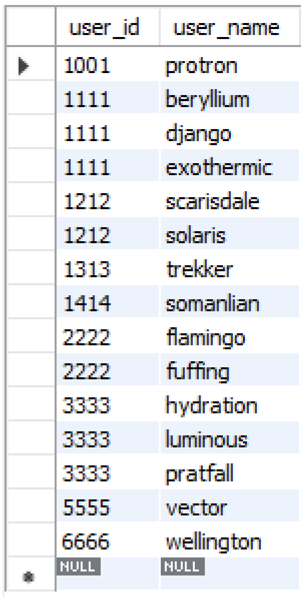
wishlist_name	car_id
Look For	1
Future Buy	2
Look For	3
Never	4
Price Check	5
When drop	6
Just looking	7
One day	8
Just a dream	9
Just looking	10
Never	11
Just a dream	12
Just looking	13
Give me	14
Honda List	11

admin_updates_car



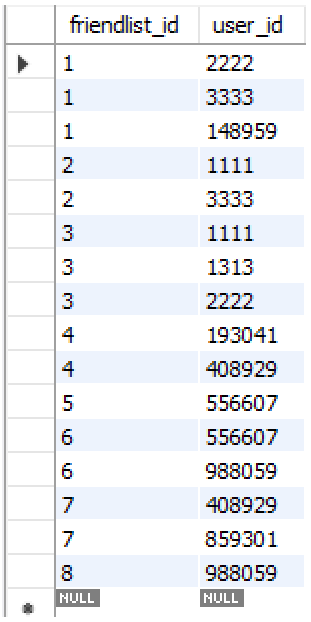
user_id	car_id
1111	1
1313	10
1515	11
1616	15
2222	2
2222	3
2222	4
2222	5
2222	6
3333	7
3333	8
3333	9
7777	12
8888	13
8888	14
NULL	NULL

admin_updates_consumer



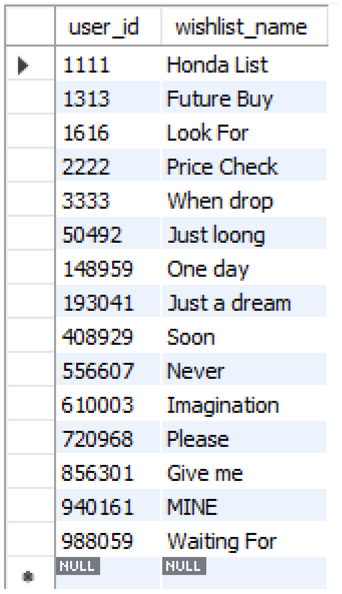
user_id	user_name
1001	protron
1111	beryllium
1111	django
1111	exothermic
1212	scarisdale
1212	solaris
1313	trekker
1414	somanlian
2222	flamingo
2222	fuffing
3333	hydration
3333	luminous
3333	pratfall
5555	vector
6666	wellington
NULL	NULL

user_has_friendlist



friendlist_id	user_id
1	2222
1	3333
1	148959
2	1111
2	3333
3	1111
3	1313
3	2222
4	193041
4	408929
5	556607
6	556607
6	988059
7	408929
7	859301
8	988059
NULL	NULL

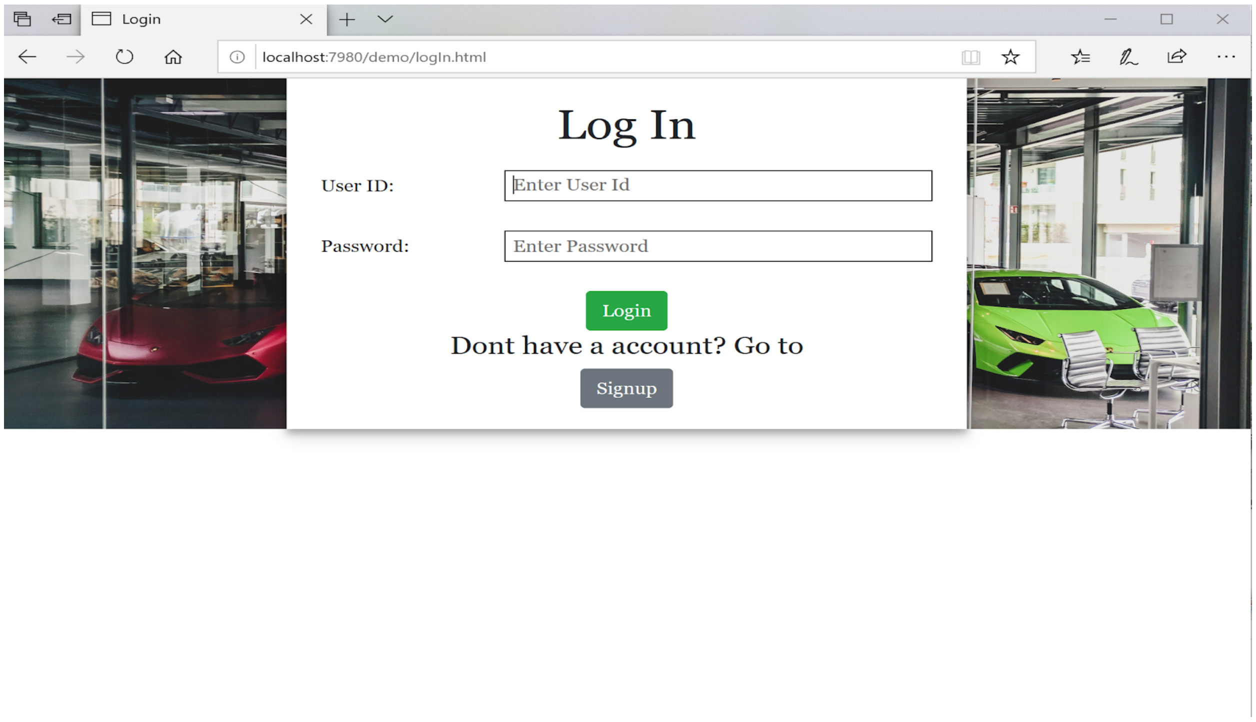
user_has_wishlist



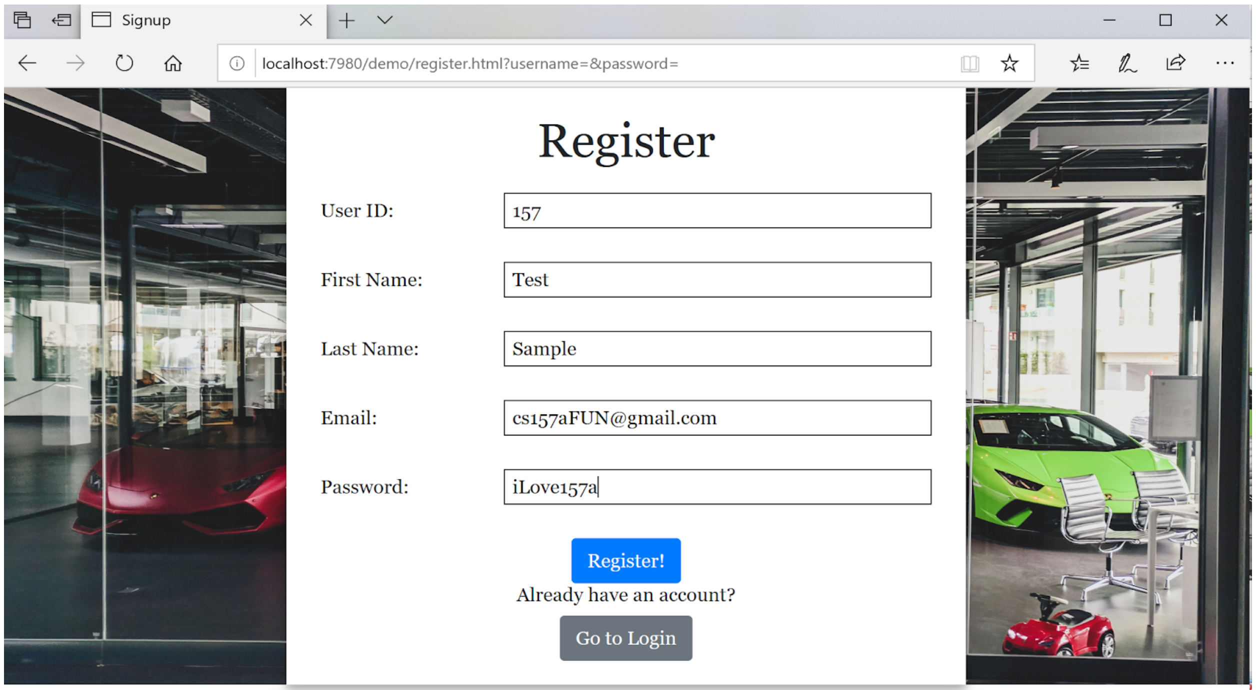
user_id	wishlist_name
1111	Honda List
1313	Future Buy
1616	Look For
2222	Price Check
3333	When drop
50492	Just loong
148959	One day
193041	Just a dream
408929	Soon
556607	Never
610003	Imagination
720968	Please
856301	Give me
940161	MINE
988059	Waiting For
NULL	NULL

Project Implementation:

User starts at Log In page to either log in, or to sign up.



Register Page:



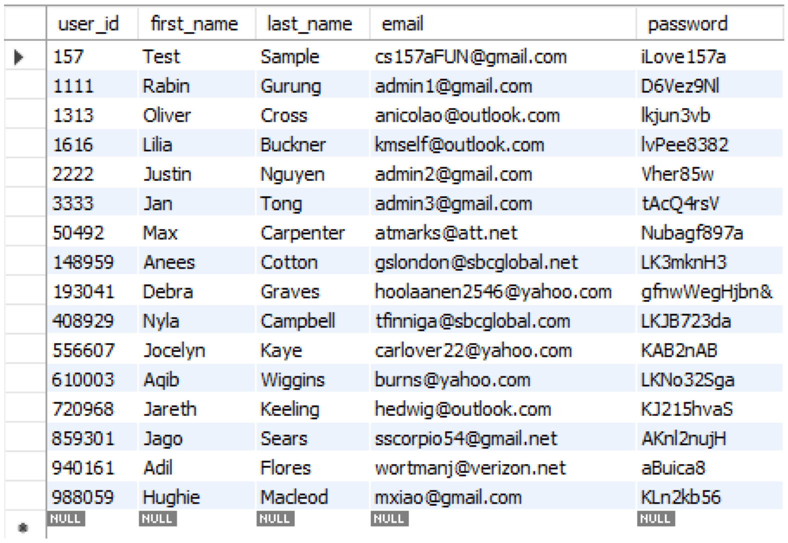
Signup

localhost:7980/demo/register.html?username=&password=

Register

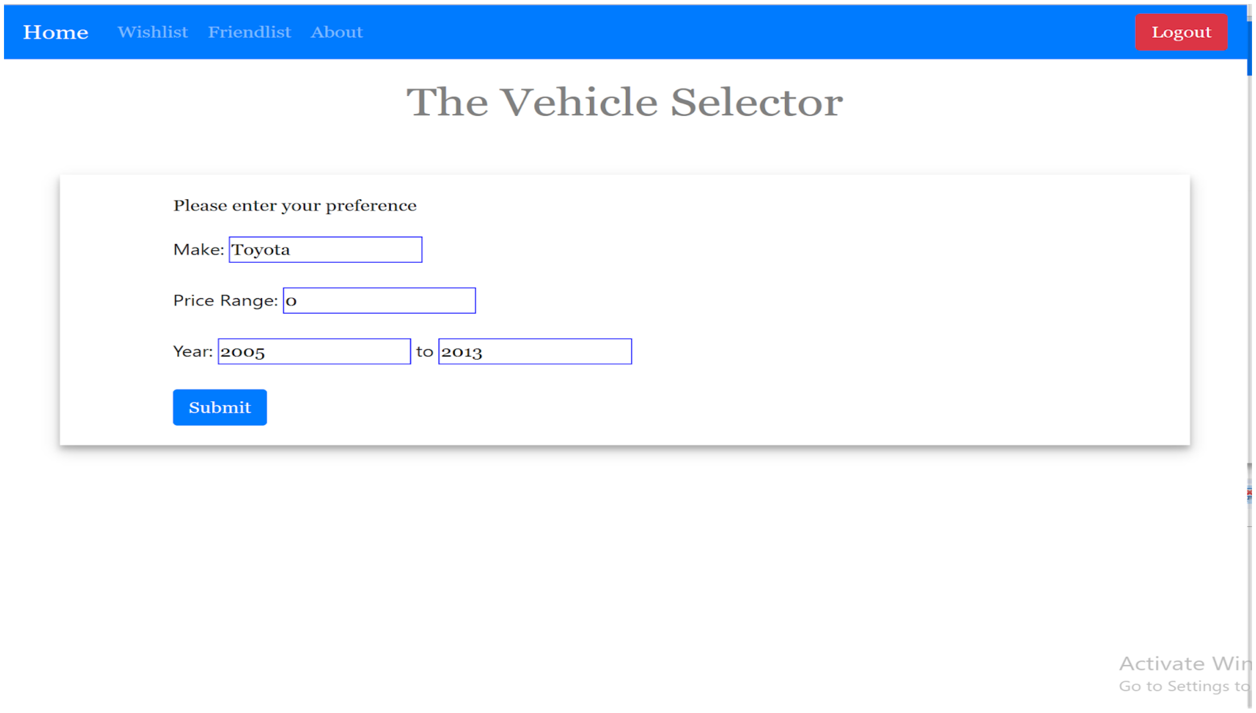
User ID:	157
First Name:	Test
Last Name:	Sample
Email:	cs157aFUN@gmail.com
Password:	iLove157a

Test subject for register is added at the first column of user table



user_id	first_name	last_name	email	password
157	Test	Sample	cs157aFUN@gmail.com	iLove157a
1111	Rabin	Gurung	admin1@gmail.com	D6Vez9Nl
1313	Oliver	Cross	anicolao@outlook.com	lkjun3vb
1616	Lilia	Buckner	kmself@outlook.com	lvPee8382
2222	Justin	Nguyen	admin2@gmail.com	Vher85w
3333	Jan	Tong	admin3@gmail.com	tAcQ4rsV
50492	Max	Carpenter	atmarks@att.net	Nubagf897a
148959	Anees	Cotton	gslondon@sbcglobal.net	LK3mknH3
193041	Debra	Graves	hoolaanen2546@yahoo.com	gfnwWegHjbn&
408929	Nyla	Campbell	tfinniga@sbcglobal.com	LKJB723da
556607	Jocelyn	Kaye	carlover22@yahoo.com	KAB2nAB
610003	Aqib	Wiggins	burns@yahoo.com	LKNo32Sga
720968	Jareth	Keeling	hedwig@outlook.com	KJ215hvaS
859301	Jago	Sears	sscorpio54@gmail.net	AKnl2nujH
940161	Adil	Flores	wortmanj@verizon.net	aBuica8
988059	Hughie	Macleod	mxiao@gmail.com	KLn2kb56
NULL	NULL	NULL	NULL	NULL

Main page after log in
In this example, a Toyota with unlimited price range from 2005 to 2013



The Vehicle Selector

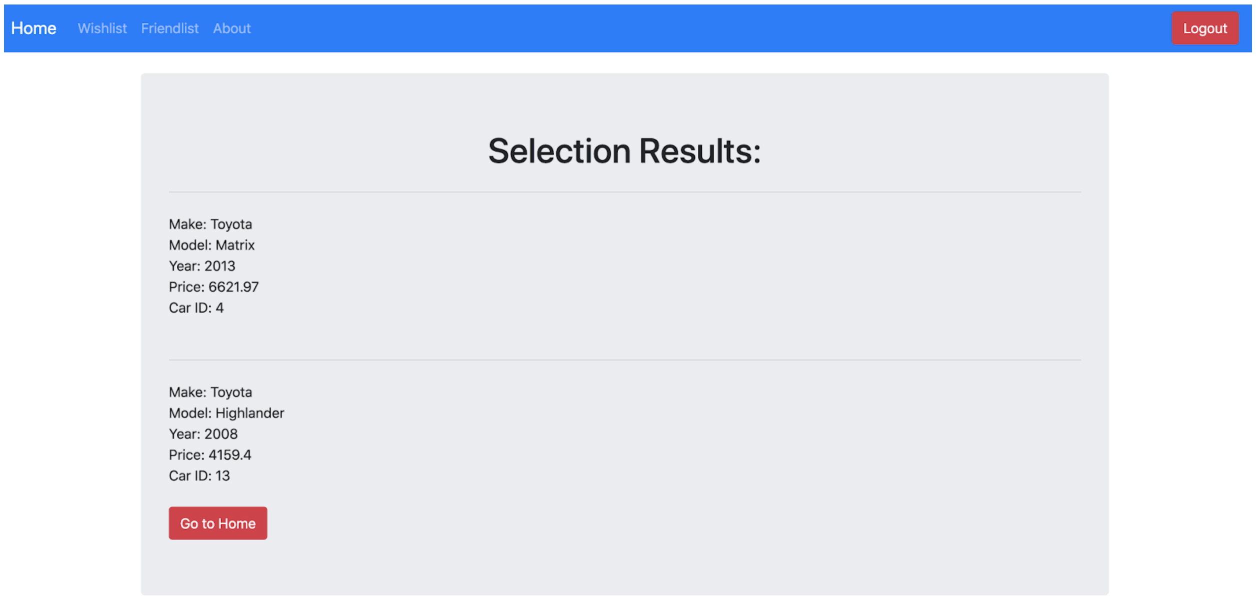
Please enter your preference

Make: Toyota

Price Range: 0

Year: 2005 to 2013

Result of searching (submit)

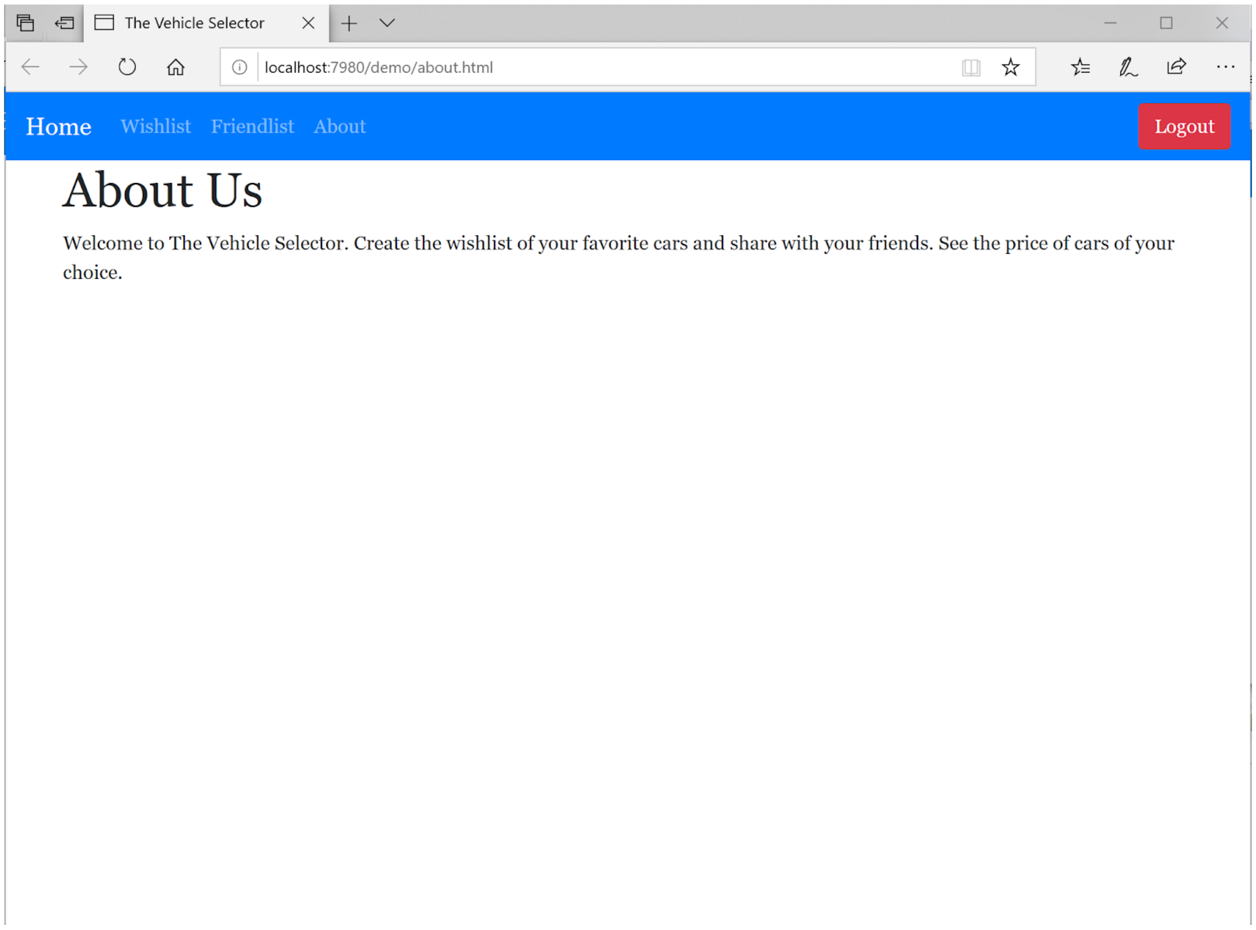


Selection Results:

Make: Toyota
Model: Matrix
Year: 2013
Price: 6621.97
Car ID: 4

Make: Toyota
Model: Highlander
Year: 2008
Price: 4159.4
Car ID: 13

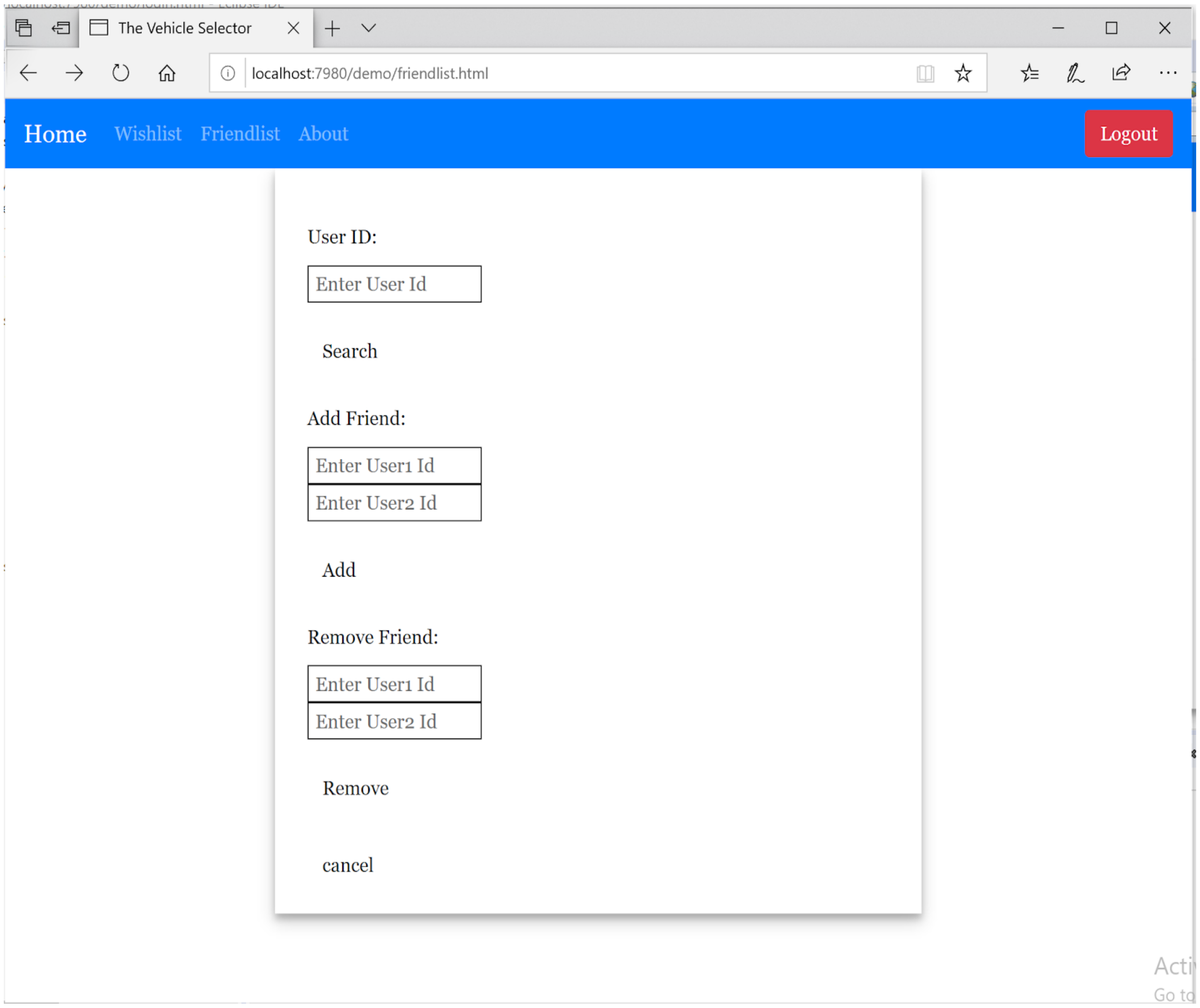
About Page (selected from the bar at the top)



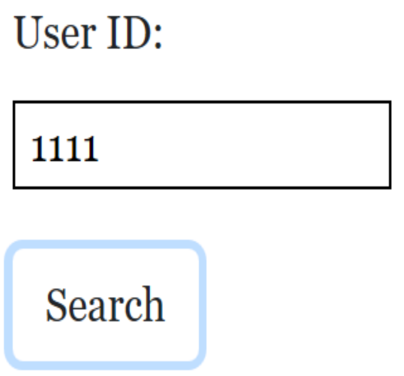
About Us

Welcome to The Vehicle Selector. Create the wishlist of your favorite cars and share with your friends. See the price of cars of your choice.

Friend List Page (selected from the bar at the top)

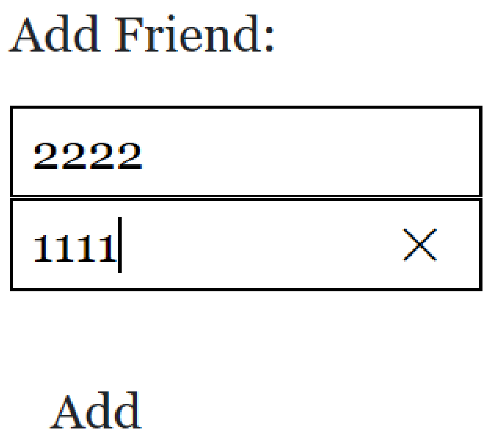


Example of inquiring friend list of User ID 1111

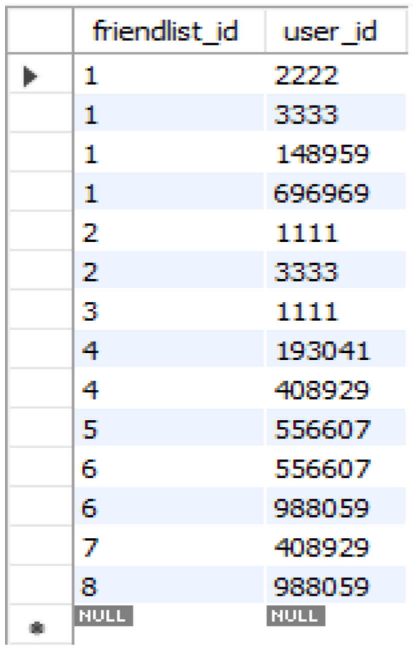


Outcome of query

Adding User 1111 to User 2222's friend list

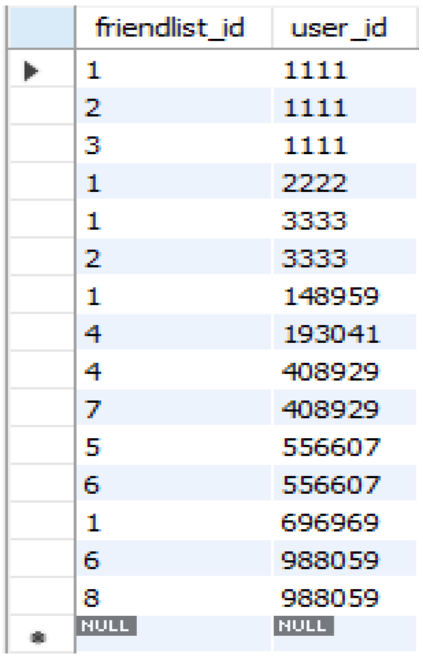


Original List



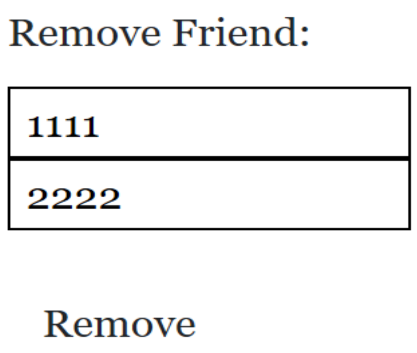
	friendlist_id	user_id
▶	1	2222
	1	3333
	1	148959
	1	696969
	2	1111
	2	3333
	3	1111
	4	193041
	4	408929
	5	556607
	6	556607
	6	988059
	7	408929
	8	988059
*	NULL	NULL

After adding friend (both 2222 and 1111 in list 1)



	friendlist_id	user_id
▶	1	1111
	2	1111
	3	1111
	1	2222
	1	3333
	2	3333
	1	148959
	4	193041
	4	408929
	7	408929
	5	556607
	6	556607
	1	696969
	6	988059
	8	988059
*	NULL	NULL

Remove User ID 2222 from User 1111's friend list

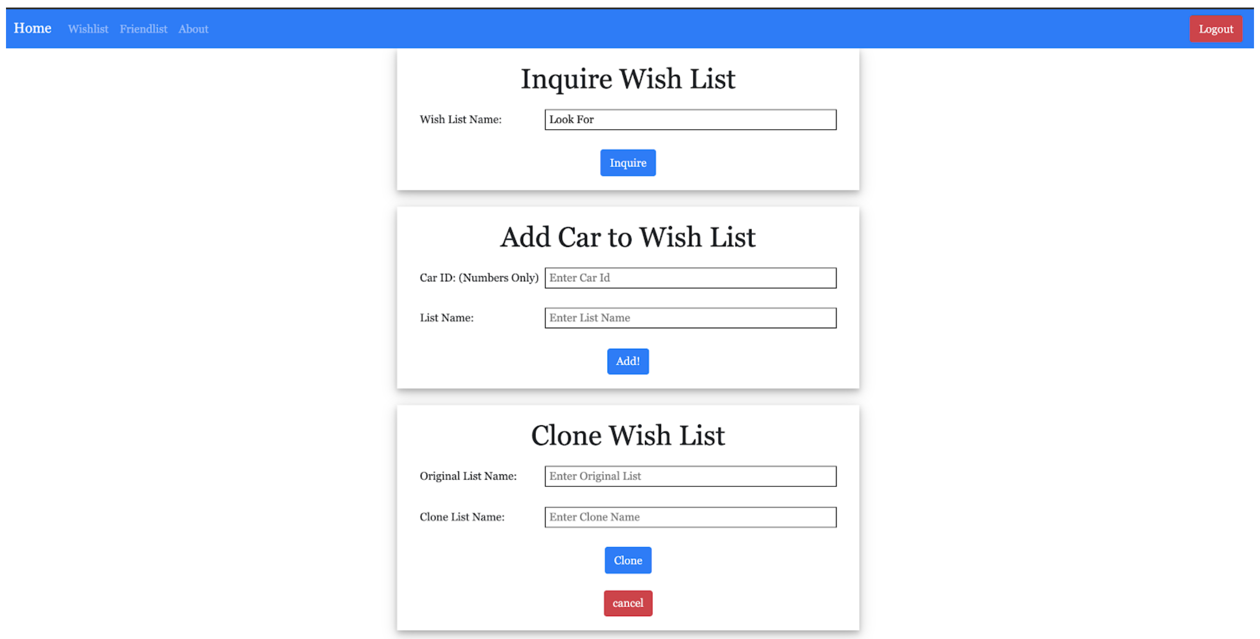


Remove Friend:

1111
2222

Remove

Wish List page (selected from the bar at the top)



Inquire Wish List

Wish List Name: Look For

Add Car to Wish List

Car ID: (Numbers Only) Enter Car Id

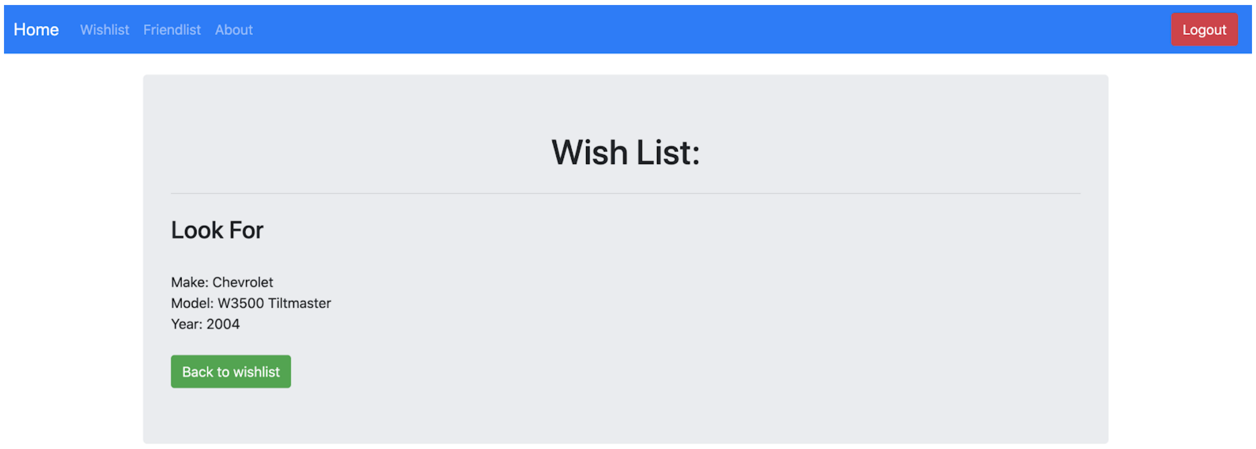
List Name: Enter List Name

Clone Wish List

Original List Name: Enter Original List

Clone List Name: Enter Clone Name

Inquiring “Look For” wish list

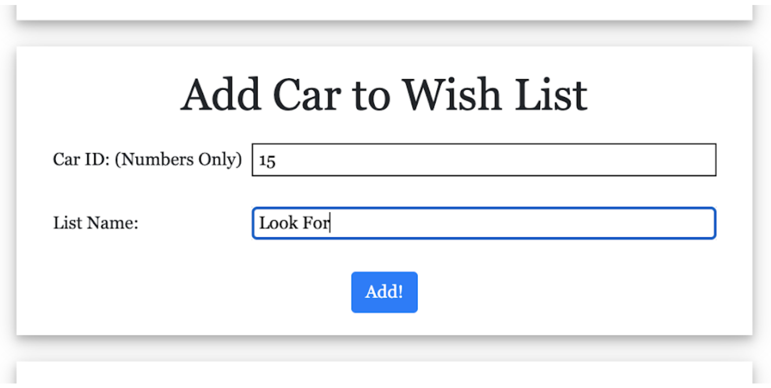


Wish List:

Look For

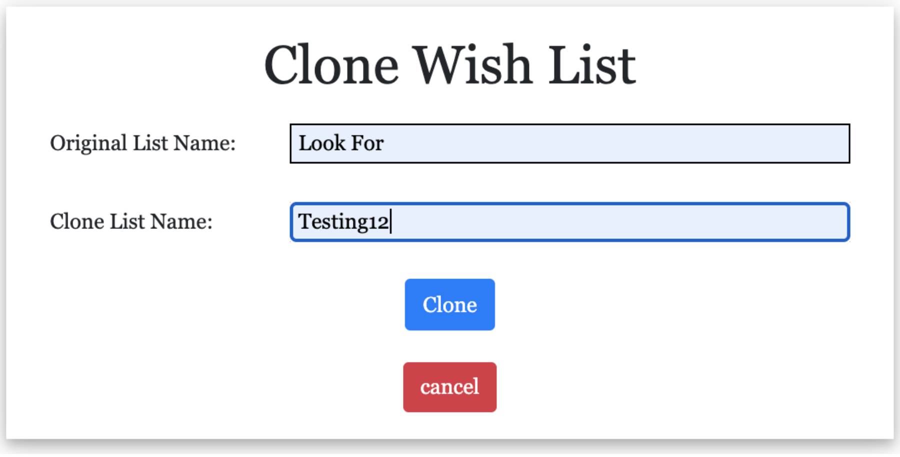
Make: Chevrolet
Model: W3500 Tiltmaster
Year: 2004

Add car to wish list option (adds car to wish list)
In this example a Mercedez(ID 15) is added to the wish list “Look For”


Add Car to Wish List

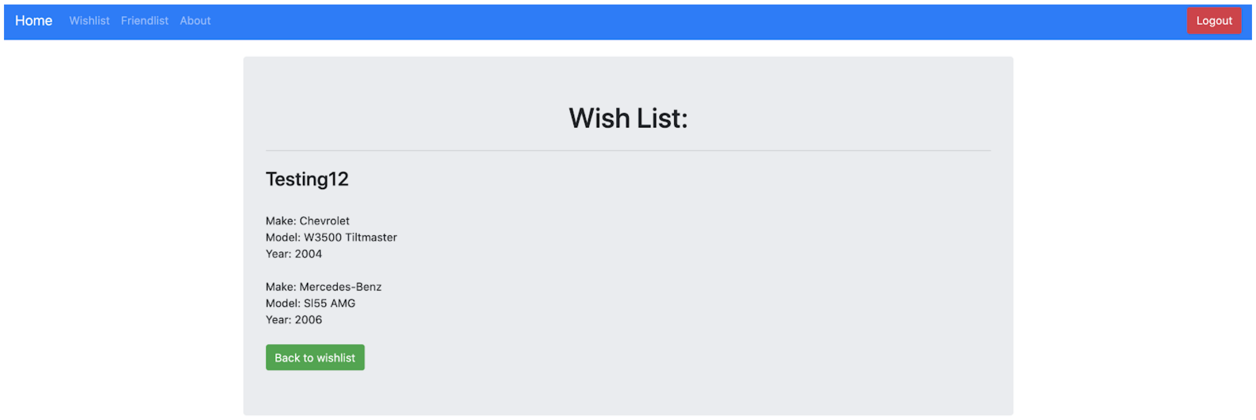
Car ID: (Numbers Only)	15
List Name:	Look For

Clone Wish list
In this example, creating clone “Testing12” from “Look For”


Clone Wish List

Original List Name:	Look For
Clone List Name:	Testing12

Resulting inquiry of “Testing12”



Home Wishlist Friendlist About Logout

Wish List:

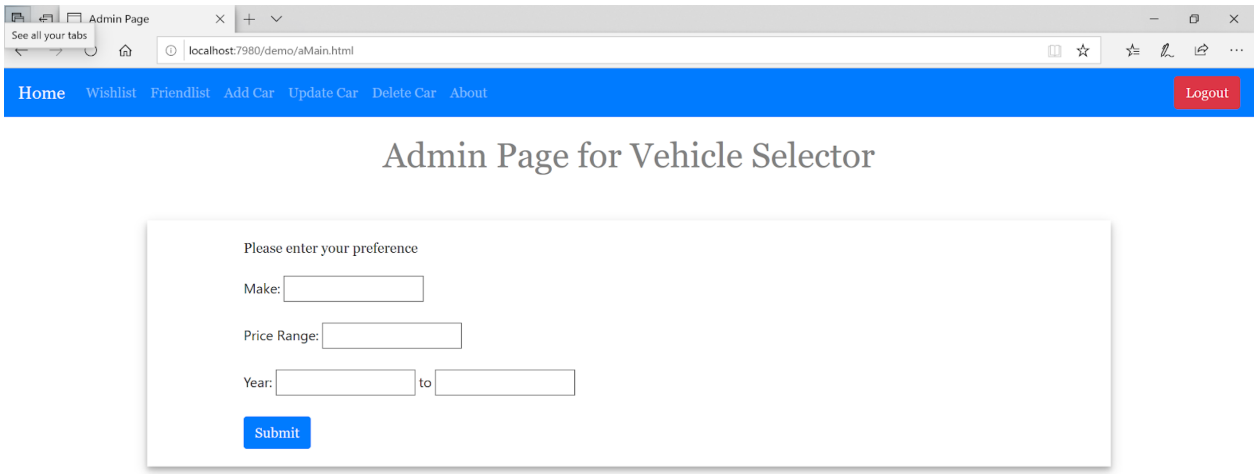
Testing12

Make: Chevrolet
Model: W3500 Tiltmaster
Year: 2004

Make: Mercedes-Benz
Model: Sl55 AMG
Year: 2006

Back to wishlist

Admin Page, after logging in with an admin account
Admin account has 3 extra options at the top bar: Add Car, Update Car, and Delete Car



Admin Page
See all your tabs
localhost:7980/demo/aMain.html

Home Wishlist Friendlist Add Car Update Car Delete Car About Logout

Admin Page for Vehicle Selector

Please enter your preference

Make:

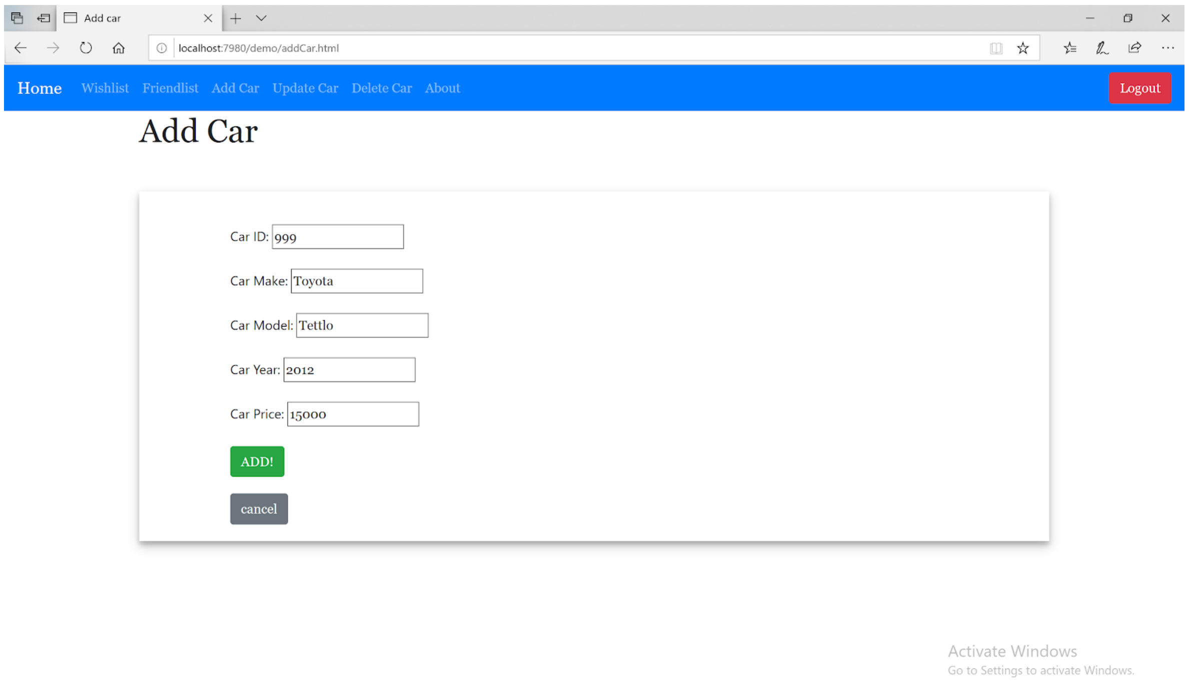
Price Range:

Year: to

Submit

Add Car page

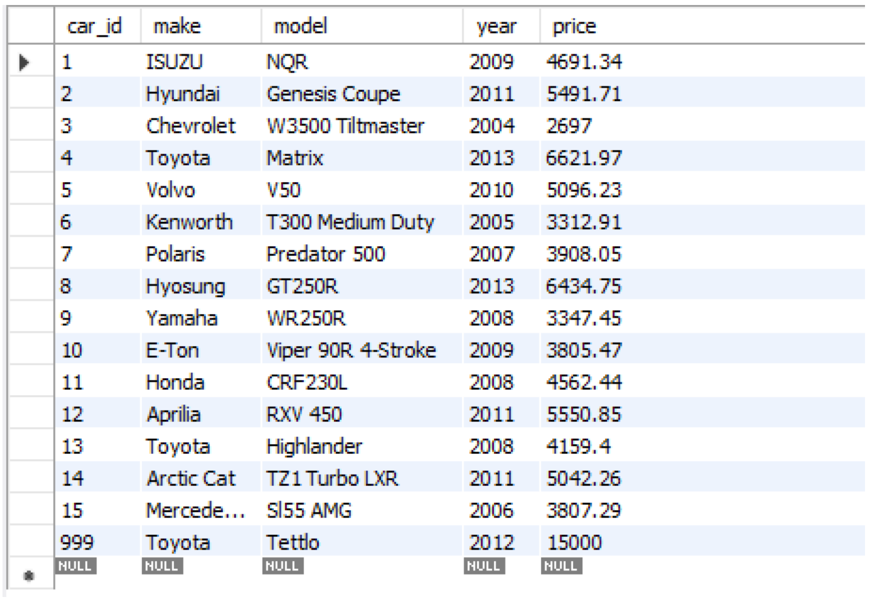
In this example, adding Toyota with car_id 999



Add Car

Car ID:	999
Car Make:	Toyota
Car Model:	Tettlo
Car Year:	2012
Car Price:	15000

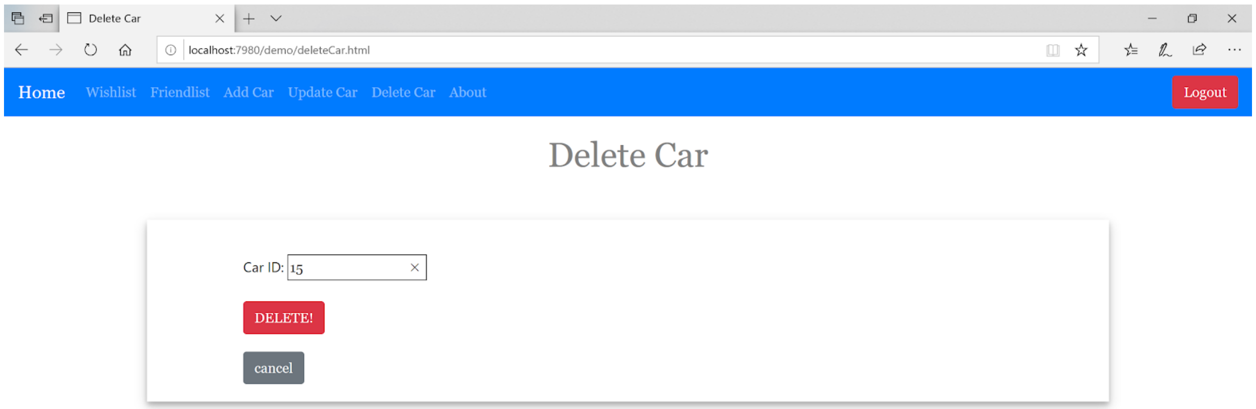
car_id 999 is found at bottom of query



car_id	make	model	year	price
1	ISUZU	NQR	2009	4691.34
2	Hyundai	Genesis Coupe	2011	5491.71
3	Chevrolet	W3500 Tiltmaster	2004	2697
4	Toyota	Matrix	2013	6621.97
5	Volvo	V50	2010	5096.23
6	Kenworth	T300 Medium Duty	2005	3312.91
7	Polaris	Predator 500	2007	3908.05
8	Hyosung	GT250R	2013	6434.75
9	Yamaha	WR250R	2008	3347.45
10	E-Ton	Viper 90R 4-Stroke	2009	3805.47
11	Honda	CRF230L	2008	4562.44
12	Aprilia	RXV 450	2011	5550.85
13	Toyota	Highlander	2008	4159.4
14	Arctic Cat	TZ1 Turbo LXR	2011	5042.26
15	Mercede...	Sl55 AMG	2006	3807.29
999	Toyota	Tettlo	2012	15000
NULL	NULL	NULL	NULL	NULL

Delete car page

In this example, removing car_id 15 from database



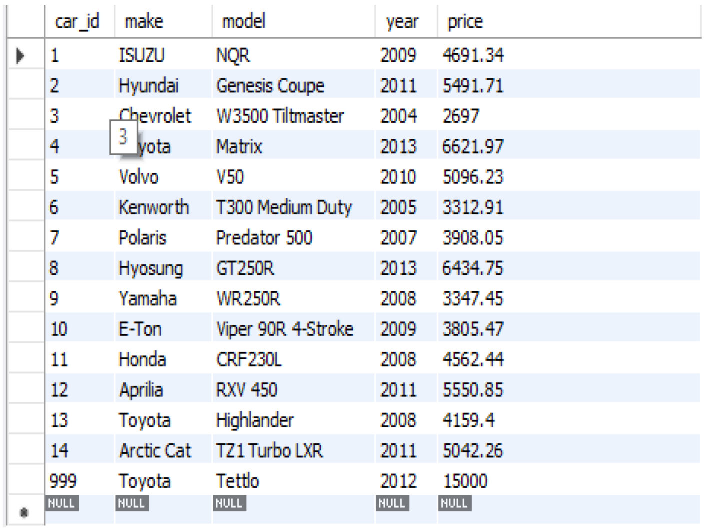
Delete Car

Car ID: 15

DELETE!

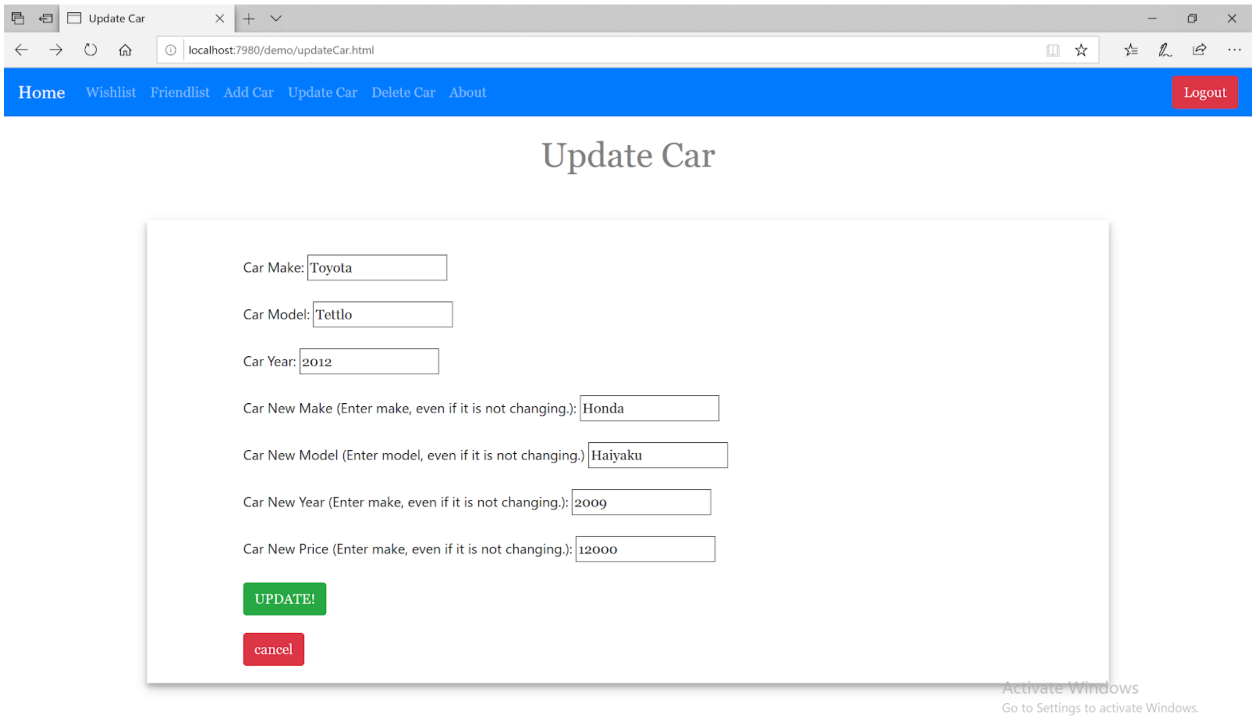
cancel

Car_id 15 removed from database



car_id	make	model	year	price
1	ISUZU	NQR	2009	4691.34
2	Hyundai	Genesis Coupe	2011	5491.71
3	Chevrolet	W3500 Tiltmaster	2004	2697
4	yota	Matrix	2013	6621.97
5	Volvo	V50	2010	5096.23
6	Kenworth	T300 Medium Duty	2005	3312.91
7	Polaris	Predator 500	2007	3908.05
8	Hyosung	GT250R	2013	6434.75
9	Yamaha	WR250R	2008	3347.45
10	E-Ton	Viper 90R 4-Stroke	2009	3805.47
11	Honda	CRF230L	2008	4562.44
12	Aprilia	RXV 450	2011	5550.85
13	Toyota	Highlander	2008	4159.4
14	Arctic Cat	TZ1 Turbo LXR	2011	5042.26
999	Toyota	Tettlo	2012	15000
NULL	NULL	NULL	NULL	NULL

Update car example
Allow to change make, model, and year of car
In this example, changing the make of car_id 999 to “Honda”



Update Car

Car Make: Toyota

Car Model: Tettlo

Car Year: 2012

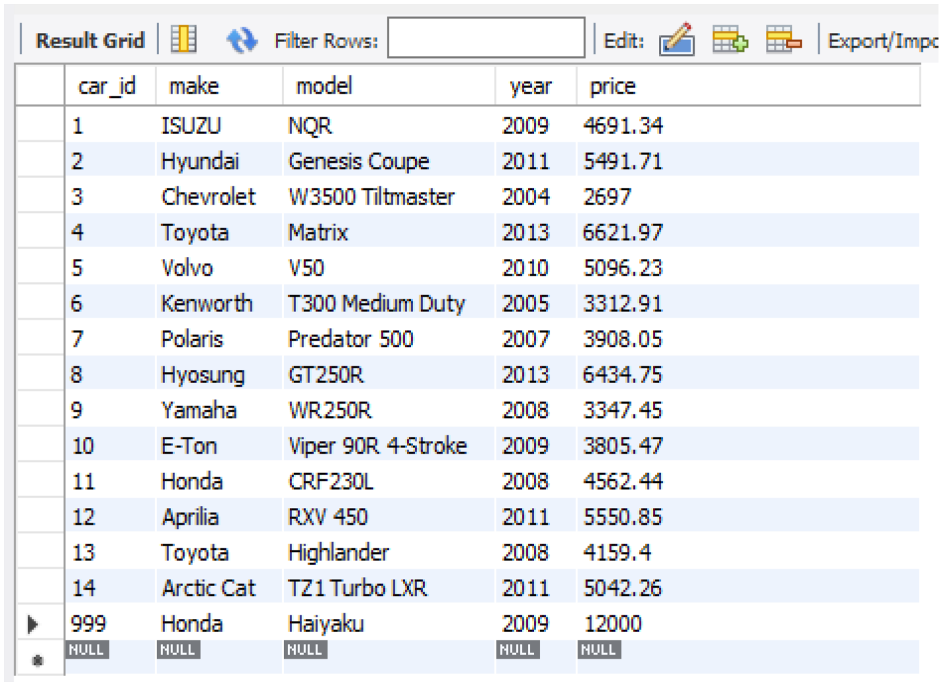
Car New Make (Enter make, even if it is not changing.): Honda

Car New Model (Enter model, even if it is not changing.) Haiyaku

Car New Year (Enter make, even if it is not changing.): 2009

Car New Price (Enter make, even if it is not changing.): 12000

Make changed to “Honda” for car_id999



car_id	make	model	year	price
1	ISUZU	NQR	2009	4691.34
2	Hyundai	Genesis Coupe	2011	5491.71
3	Chevrolet	W3500 Tiltmaster	2004	2697
4	Toyota	Matrix	2013	6621.97
5	Volvo	V50	2010	5096.23
6	Kenworth	T300 Medium Duty	2005	3312.91
7	Polaris	Predator 500	2007	3908.05
8	Hyosung	GT250R	2013	6434.75
9	Yamaha	WR250R	2008	3347.45
10	E-Ton	Viper 90R 4-Stroke	2009	3805.47
11	Honda	CRF230L	2008	4562.44
12	Aprilia	RXV 450	2011	5550.85
13	Toyota	Highlander	2008	4159.4
14	Arctic Cat	TZ1 Turbo LXR	2011	5042.26
999	Honda	Haiyaku	2009	12000
NULL	NULL	NULL	NULL	NULL

Take Away:

Jan To Tong:
Through this project, I was able to understand how a 3-tier architecture system works, where I had the opportunity to learn how a front-end system interacts with the back-end while working in a team environment. I was able to learn front-end languages (HTML5, CSS) to create the foundations for a website, XML mapping to connect the Java servlets with the front-end, connecting the Java servlets with the RDBMS (MySQL) to update or query the database, and hosting a web server using Apache Tomcat. Overall, this project taught me about how a web application interacts with its database system, and how to distribute work within a coding team even at times where we couldn’t physically meet.

Justin Nguyen:
While working on this project, I was able to learn many different things. First being able to understand how mysql works, and being able to connect mysql to a java code. I also learned how to use multiple query statements to edit the database without directly using mysql workbench. The project required me to learn how to write html code and my teammates taught me how to design better html by using css. By working on this project with my teammates I was able to learn a lot of new things relating to database, website development, and working together as a team from home.

Rabin Gurung:
During this semester, I learned how to access and manipulate the databases. I also learned lots of logics or ideas to manipulate the databases in an easier and convenient way. This project helped me to understand how we can connect these databases to the backend and portray in the frontend. I never worked with other colleagues in a virtual platform through github before, and I got the great opportunity to learn to work in a team. Now, I have more ideas to create a better UI design using HTML and CSS than before. I have learned to write a code to connect the Backend to FrontEnd and MySQL.