COMP 306 TERM PROJECT

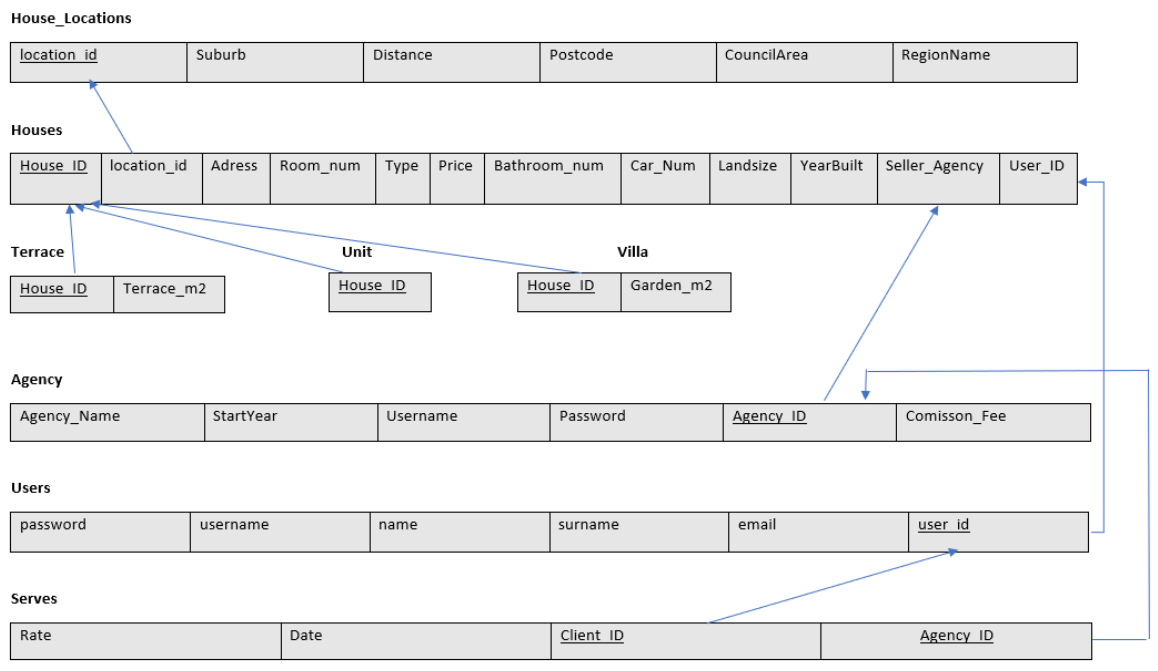
MELBOURNE REAL STATE AGENCIES

İrem Nur Bulut -68776
Ece Güz - 69002
Zeynep Sıla Kaya - 69101
Doğa Demirtürk - 68859

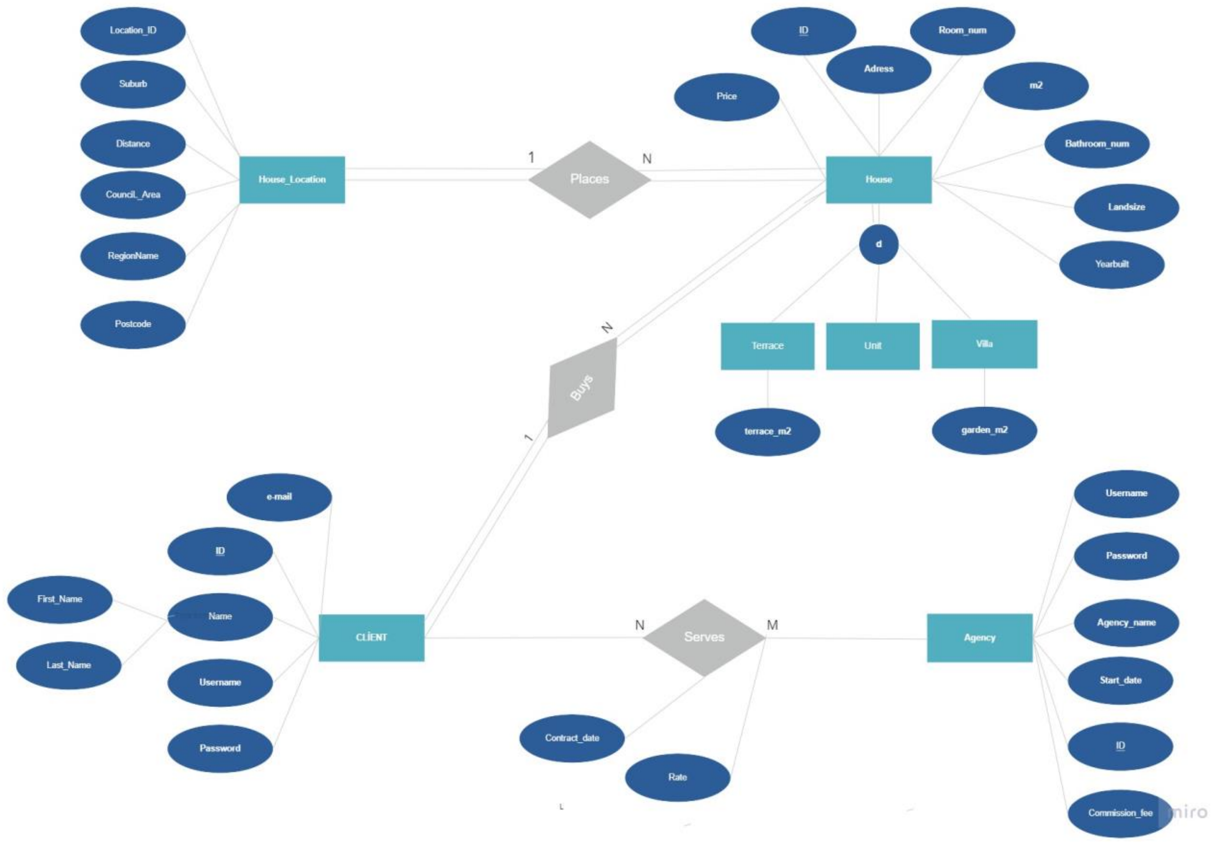
1) Textual description of our project:

We created a Real-Estate Agency management system. Our system currently contains thirty real estate agencies and it includes about 500 customer users. In addition, new users can become a member of the system. When the program starts, you see the login page. If you have a user account, you can log in as a client or as an agency. If you are a customer and do not have a user account, you can become a member by clicking the Sign In button. When you log in to the site, if you are an agency, you will see the agency home page. On the agency homepage, agencies can add a new house for sale, remove an existing house (for example, if the landlord decides to not sell the house), view their ranking by seeing the votes of the agencies given by users, and see the average income of each agency from the houses sold so far. By clicking the logout button, the agency account can exit from the system and return to the login page. On the customer home page, they can see the houses for sale by selecting the criteria they want. When they find the house they want, they can buy it by pressing the buy button. By clicking the Agencies button, they can sort the agencies according to the criteria they want. They can log out of their accounts by clicking the logout button and return to the login page.

2) Relational database schema:



Additional: ER Diagram for better Visualisation



3) Create Table Statements:

```
CREATE TABLE `agencies` (
  `Agency_Name` varchar(13) DEFAULT NULL,
  `StartYear` int DEFAULT NULL,
  `Username` varchar(18) CHARACTER SET utf8 COLLATE utf8_general_ci DEFAULT NULL,
  `Password` varchar(8) CHARACTER SET utf8 COLLATE utf8_general_ci DEFAULT NULL,
  `Agency_ID` int NOT NULL,
  `Commision_fee` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
ALTER TABLE `agencies`
  ADD PRIMARY KEY (`Agency_ID`),
  ADD UNIQUE KEY `Username` (`Username`);
```

```
CREATE TABLE `house_locations` (
  `location_id` int NOT NULL,
  `Suburb` varchar(50) NOT NULL,
  `Distance` int NOT NULL,
  `Postcode` int NOT NULL,
  `CouncilArea` varchar(50) NOT NULL,
  `RegionName` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
ALTER TABLE `house_locations`
  ADD PRIMARY KEY (`location_id`);
```

```
CREATE TABLE `houses` (
  `Address` varchar(27) CHARACTER SET utf8 COLLATE utf8_general_ci DEFAULT NULL,
  `RoomNum` int DEFAULT NULL,
  `Type` varchar(1) CHARACTER SET utf8 COLLATE utf8_general_ci DEFAULT NULL,
  `Price` int DEFAULT NULL,
  `Seller_Agency` int DEFAULT NULL,
  `Bathroom_Num` int DEFAULT NULL,
  `Car_Num` int DEFAULT NULL,
  `Landsize` int DEFAULT NULL,
  `YearBuilt` int DEFAULT NULL,
  `House_ID` int NOT NULL,
  `Location_ID` int DEFAULT NULL,
  `User_ID` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

ALTER TABLE `houses`
  ADD PRIMARY KEY (`House_ID`),
  ADD KEY `houses_ibfk_1` (`Location_ID`);

ALTER TABLE `houses`
  ADD CONSTRAINT `houses_ibfk_1` FOREIGN KEY (`Location_ID`) REFERENCES
`house_locations` (`location_id`) ON DELETE CASCADE ON UPDATE RESTRICT;
```

```
CREATE TABLE `villas_wc` (
  `House_ID` int NOT NULL,
  `Garden_m2` int NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
ALTER TABLE `villas_wc`
  ADD PRIMARY KEY (`House_ID`) USING BTREE;
ALTER TABLE `villas_wc`
  ADD CONSTRAINT `villas_ibfk_1` FOREIGN KEY (`House_ID`) REFERENCES `houses`
(`House_ID`) ON DELETE CASCADE ON UPDATE RESTRICT;

---

CREATE TABLE `serves` (
  `Agency_ID` int DEFAULT NULL,
  `Date` varchar(10) CHARACTER SET utf8 COLLATE utf8_general_ci DEFAULT NULL,
  `Client_ID` int DEFAULT NULL,
  `Rate` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
ALTER TABLE `serves`
  ADD KEY `serves_ibfk_1` (`Agency_ID`);
ALTER TABLE `serves`
  ADD CONSTRAINT `serves_ibfk_1` FOREIGN KEY (`Agency_ID`) REFERENCES
`agencies` (`Agency_ID`) ON DELETE CASCADE ON UPDATE RESTRICT;


---
CREATE TABLE `terrace_wc` (
  `House_ID` int NOT NULL,
  `Terrace_m2` int NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
ALTER TABLE `terrace_wc`
  ADD PRIMARY KEY (`House_ID`) USING BTREE;
ALTER TABLE `terrace_wc`
  ADD CONSTRAINT `terrace_ibfk_1` FOREIGN KEY (`House_ID`) REFERENCES `houses`
(`House_ID`) ON DELETE CASCADE ON UPDATE RESTRICT;

---

CREATE TABLE `units` (
  `House_ID` int NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
ALTER TABLE `units`
  ADD KEY `units_ibfk_1` (`House_ID`);
ALTER TABLE `units`
  ADD CONSTRAINT `unit_ibfk_1` FOREIGN KEY (`House_ID`) REFERENCES `houses`
(`House_ID`) ON DELETE CASCADE ON UPDATE RESTRICT;

---

CREATE TABLE `users` (
  `user_id` int NOT NULL,
  `username` varchar(15) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci
DEFAULT NULL,
  `name` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  `surname` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci
DEFAULT NULL,
  `email` varchar(30) DEFAULT NULL,
  `password` varchar(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
ALTER TABLE `users`
  ADD PRIMARY KEY (`user_id`),
  ADD UNIQUE KEY `email` (`email`),
  ADD UNIQUE KEY `username` (`username`);
```

4) Data Source:

We found our two datasets in Kaggle and we tried to make the datasets we found meet our requests in the best way. Firstly, we found data which includes house information of Melbourne/Australia. When we examined the Data Set, we noticed a column named house type. We thought that we can create separate tables with this house type column as villa, terrace and unit house.Thus, we have obtained the superclass subclass relation in our database.Then we saw that the house table is still very large and the columns Suburb, Distance, Postcode, Council Area Region Name repeat for each house in the same neighborhood so there was a redundancy in the table. So we created a house location table to avoid these repeating rows. We added location_id to this table and we added this as foreign key for each house to the house table. We also found the data in the Users table from Kaggle but this table only contained username and password. That's why we created an email, name and surname for everyone by using their usernames in the table. Finally, we randomly generated the data in the Agencies table.

5) 5 complex SQL queries

We have constructed our queries in the application to apply useful statistics and retrieve related information about the agencies and the houses. Below we have listed why they are useful for our system and where they have been integrated into.

Query 1)

```
SELECT agencies.Agency_ID,agencies.Agency_Name,
AVG(houses.Price*agencies.Commision_fee/100)
FROM houses JOIN agencies ON agencies.Agency_Name=houses.Seller_Agency
Where houses.User_ID NOT IN (SELECT User_ID FROM HOUSES WHERE User_ID='')
Group By agencies.Agency_ID, agencies.Agency_Name ORDER BY agencies.Agency_ID ASC
```

Our first query calculates the average commission fee that the agencies have gotten from the houses they have sold. It represents the results in ascending order. This by query retrieved information is important for the agencies to be able to compare themselves with the market average and take the appropriate actions for it. For example an agency seeing that they have a higher commission fee than the average could aim for the sale of luxury houses or could lower the prices to maximize the revenue. It takes place in the main agency page of the prototype.

Query 2)

```
SELECT agencies.Agency_ID,Agency_Name, AVG(Rate) FROM agencies JOIN serves ON
agencies.Agency_ID=serves.Agency_ID
GROUP BY  agencies.Agency_ID,agencies.Agency_Name
ORDER BY agencies.Agency_ID ASC
```

Our second query calculates the average rate that the agencies have gotten from the clients, who they have served and it represents the results in ascending order. This information is crucial for both the agencies and the clients, because it helps the clients to choose a better service while motivating the agencies to provide a better service. Moreover since the agencies with worse ratings have a generally lower revenue due to the lack of clients, the entry and exit to the market is frequent and it helps our application to be active and efficient. This query takes place in the main agency page of the prototype.

Query 3)

```
SELECT < h_type >.House_ID AS House_ID, house_locations.RegionName,
house_locations.CouncilArea, house_locations.Suburb, house_locations.Distance
FROM houses,house_locations, < h_type>
WHERE <h_type>.House_ID=houses.House_ID AND
houses.Location_ID=house_locations.Location_ID AND  < h_type>.House_ID IN (SELECT
houses.House_ID FROM houses, house_locations where
houses.Location_ID=house_locations.location_id AND
house_locations.CouncilArea=<councilArea>)
ORDER BY Distance ASC
```

Our third query sorts houses of a selected type such as villa, house or unit in the council area selected and also gives the distance of them to the center. This complex query is necessary for the clients to understand if the house can be appropriate for their needs. After this sort they can look for the list of houses suitable for their needs and decide if the distance to the center is appropriate or not. It takes place in the main client page of the prototype.

Query 4)

```
SELECT T.average FROM (SELECT A1.Username, AVG(H1.Price * A1.Commision_fee / 100)
AS average FROM houses AS H1 JOIN agencies AS A1 ON A1.Agency_ID=H1.Seller_Agency
WHERE H1.User_ID NOT IN (SELECT H2.House_ID FROM houses AS H2 WHERE
H2.User_ID=0) GROUP BY A1.Username HAVING A1.Username=<username>) AS T WHERE
T.average > (SELECT AVG(H3.Price * A3.Commision_fee / 100) FROM houses AS H3 JOIN
agencies as A3 ON A3.Agency_ID=H3.Seller_Agency Where H3.User_ID NOT IN (SELECT
H4.User_ID FROM houses AS H4 WHERE H4.User_ID=0))
```

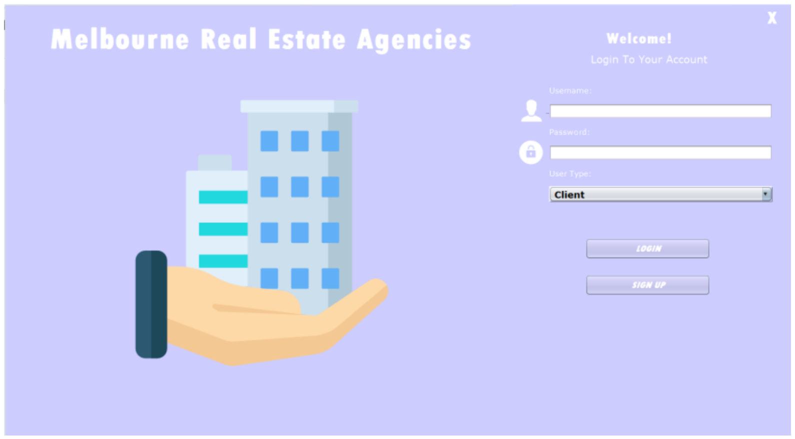
Our fourth query calculates the average revenue for the agency, which has entered the system, and compares it with the general average revenue. This query is complementary to the first query and we have developed it to further enhance the business statistics of the agencies. With the help of this query the firms are able to analyze their market cap and furthermore they can develop their revenue maximization strategy considering the general situation of their competitors. This query takes place in the main agency page of the prototype.

Query 5)

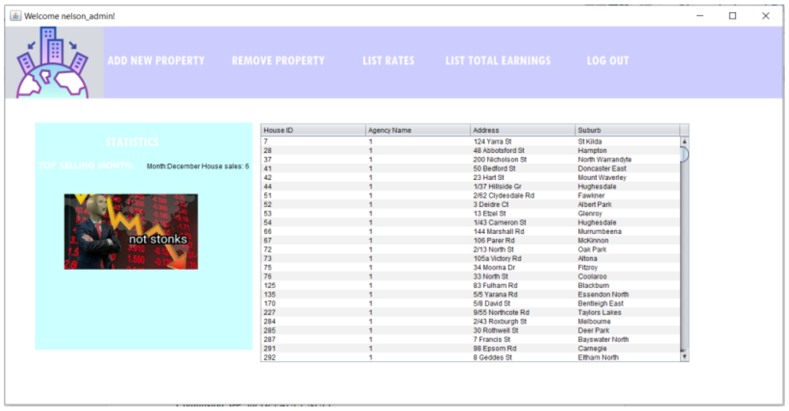
SELECT MAX(subquery.count)as number,subquery.bestMonth as month FROM(SELECT COUNT(*)as count,month(serves.Date) as bestMonth
 FROM serves,agencies WHERE serves.Agency_ID=agencies.Agency_ID AND agencies.Username= <username> GROUP by month(serves.date))as subquery

Our last query chooses the month that the agency, which has logged-in the application, has the highest sales and also shows the sale amount. This helps the agencies to orient their staff's vacation times and they can foresee the influx of accounting and tax related problems better. This query takes place in thein the main agency page of the prototype.

(6) Screenshots from our final system prototype:



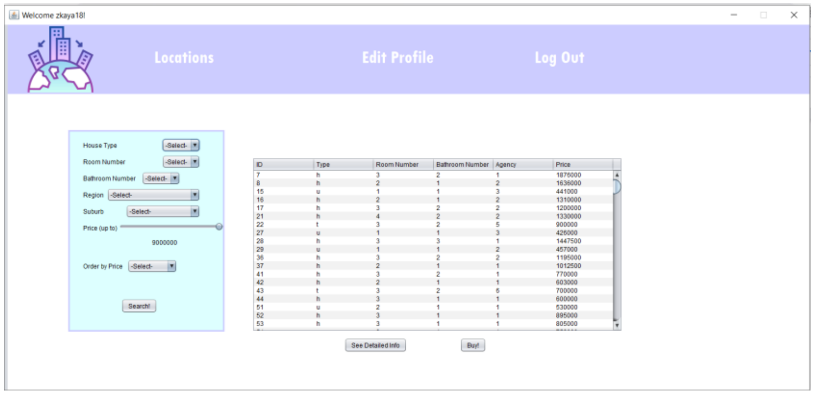
→ login page



Month:December House sales: 6

House ID	Agency Name	Address	Suburb
7	1	124 Yarra St	St Kilda
28	1	48 Abbotsford St	Hampton
37	1	200 Nicholson St	North Warrandyte
41	1	50 Bedford St	Doncaster East
42	1	23 Hart St	Mount Waverley
44	1	1/37 Hillside Gr	Hughesdale
51	1	2/62 Clydesdale Rd	Fawkner
52	1	3 Deidre Ct	Albert Park
53	1	13 Etzel St	Glenroy
54	1	1/43 Cameron St	Hughesdale
66	1	144 Marshall Rd	Murrumbeena
67	1	106 Parer Rd	McKinnon
72	1	2/13 North St	Oak Park
73	1	105a Victory Rd	Altona
75	1	34 Moorna Dr	Fitzroy
76	1	33 North St	Coolaroo
125	1	83 Fulham Rd	Blackburn
135	1	5/5 Yarana Rd	Essendon North
170	1	5/8 David St	Bentleigh East
227	1	9/55 Northcote Rd	Taylors Lakes
284	1	2/43 Roxburgh St	Melbourne
285	1	30 Rothwell St	Deer Park
287	1	7 Francis St	Bayswater North
291	1	98 Epsom Rd	Carnegie
292	1	8 Geddes St	Eltham North

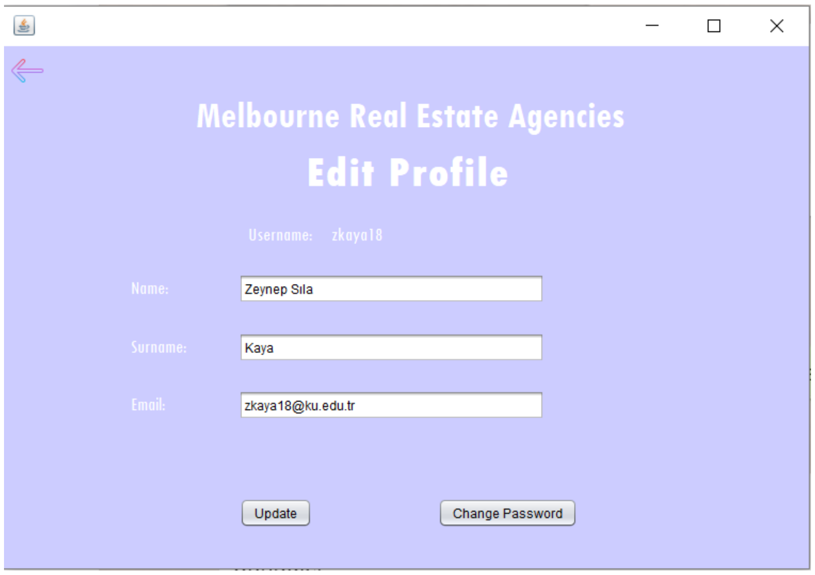
→ main agency page



House Type: -Select-
Room Number: -Select-
Bathroom Number: -Select-
Region: -Select-
Suburb: -Select-
Price (up to): 9000000
Order by Price: -Select-

ID	Type	Room Number	Bathroom Number	Agency	Price
7	h	3	2	1	1876000
8	h	2	1	2	1636000
15	u	1	1	3	441000
16	h	2	1	2	1310000
17	h	3	2	2	1200000
21	h	4	2	2	1330000
22	t	3	2	5	900000
27	u	1	1	3	426000
28	h	3	3	1	1447500
29	u	1	1	2	457000
36	h	3	2	2	1195000
37	h	2	1	1	1012500
41	h	3	2	1	770000
42	h	2	1	1	603000
43	t	3	2	6	700000
44	h	3	1	1	600000
51	u	2	1	1	530000
52	h	3	1	1	895000
53	h	3	1	1	805000

→ main client page



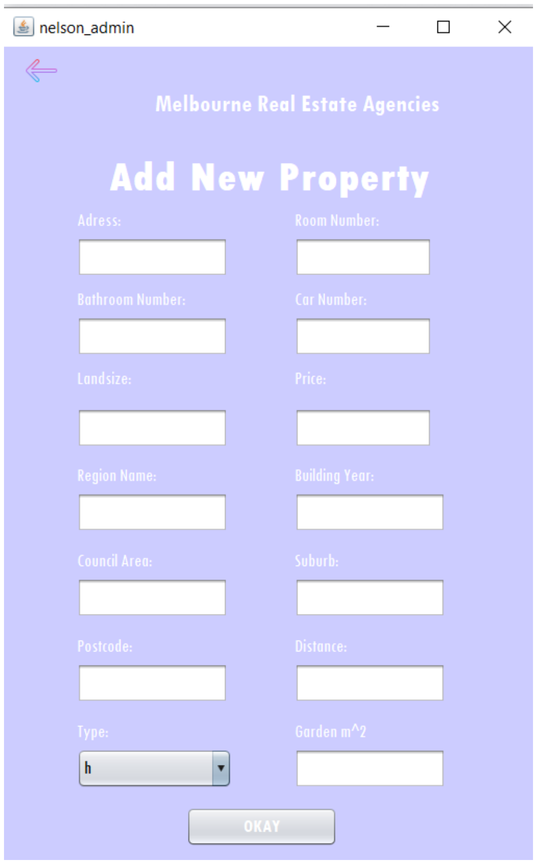
Username: zkaya18

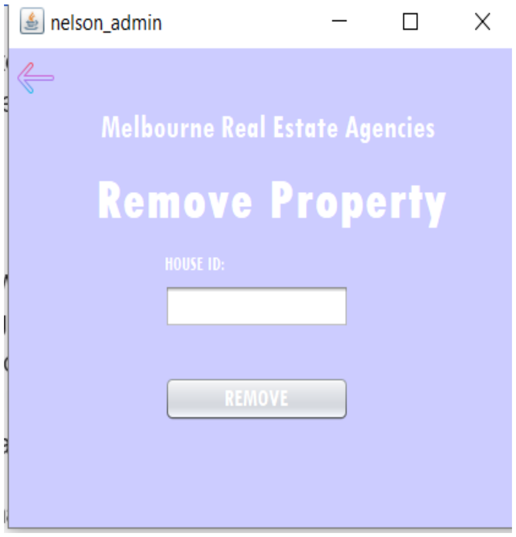
Name: Zeynep Sıla

Surname: Kaya

Email: zkaya18@ku.edu.tr

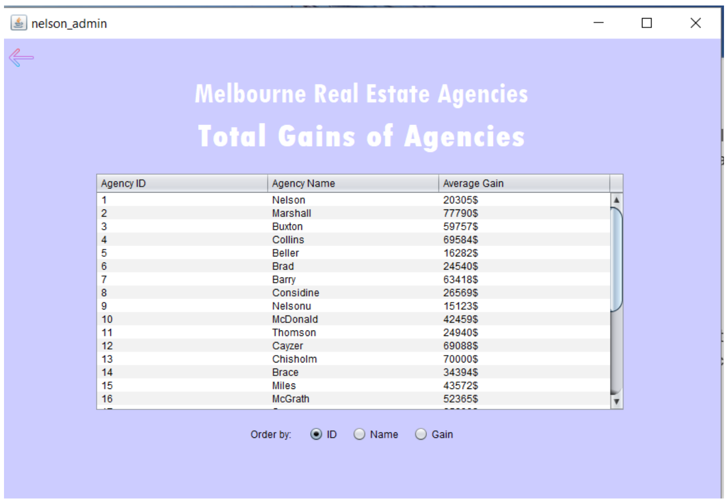
→ profile edit page





→ add new property

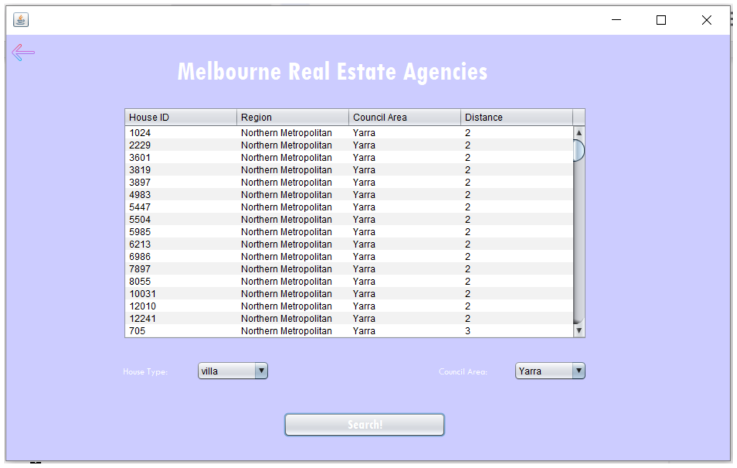
→ remove property



Agency ID	Agency Name	Average Gain
1	Nelson	20305$
2	Marshall	77790$
3	Buxton	59757$
4	Collins	69584$
5	Beller	16282$
6	Brad	24540$
7	Barry	63418$
8	Considine	26569$
9	Nelsonu	15123$
10	McDonald	42459$
11	Thomson	24940$
12	Cayzer	69088$
13	Chisholm	70000$
14	Brace	34394$
15	Miles	43572$
16	McGrath	52365$

Order by: [✓ ID] Name Gain

→ Total Gains of Agencies Page

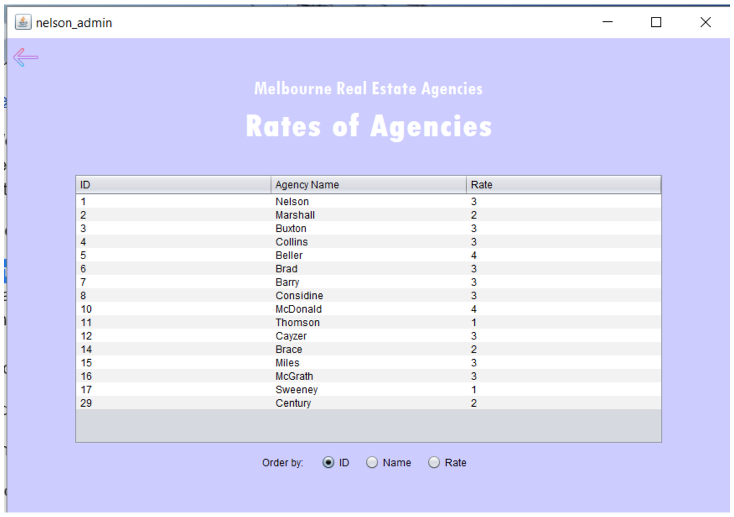


House ID	Region	Council Area	Distance
1024	Northern Metropolitan	Yarra	2
2229	Northern Metropolitan	Yarra	2
3601	Northern Metropolitan	Yarra	2
3819	Northern Metropolitan	Yarra	2
3897	Northern Metropolitan	Yarra	2
4983	Northern Metropolitan	Yarra	2
5447	Northern Metropolitan	Yarra	2
5504	Northern Metropolitan	Yarra	2
5985	Northern Metropolitan	Yarra	2
6213	Northern Metropolitan	Yarra	2
6986	Northern Metropolitan	Yarra	2
7897	Northern Metropolitan	Yarra	2
8055	Northern Metropolitan	Yarra	2
10031	Northern Metropolitan	Yarra	2
12010	Northern Metropolitan	Yarra	2
12241	Northern Metropolitan	Yarra	2
705	Northern Metropolitan	Yarra	3

House Type: villa

Council Area: Yarra

→ Main Locations & Distance Page



ID	Agency Name	Rate
1	Nelson	3
2	Marshall	2
3	Buxton	3
4	Collins	3
5	Beller	4
6	Brad	3
7	Barry	3
8	Considine	3
10	McDonald	4
11	Thomson	1
12	Cayzer	3
14	Brace	2
15	Miles	3
16	McGrath	3
17	Sweeney	1
29	Century	2

Order by: [✓ ID] Name Rate

→ Rates of Agencies Page