

# Final Report

## University Marketplace

Group 16

Mihir Kakadiya  
Vinaskumar Sukhadiya

[kakadiya.m@northeastern.edu](mailto:kakadiya.m@northeastern.edu)  
[sukhadiya.v@northeastern.edu](mailto:sukhadiya.v@northeastern.edu)

**Percentage of Effort Contributed by Student 1: 50%**

**Percentage of Effort Contributed by Student 2: 50%**

**Signature of Student 1: Mihir kakadiya**

**Signature of Student 2: Vinaskumar Sukhadiya**

**Submission Date: 04/23/2023**

## **OVERVIEW AND INTRODUCTION**

As a student, managing finances can be a challenging task. Many students often have to sell their items to make ends meet. Our project aims to create an online marketplace where students from any university can easily sell their items to other students within the same university. This platform will connect senior students who are graduating and moving out with fellow students who are in need of household items. By providing a platform that is specific to the university, students can feel more secure about buying and selling items with their peers.

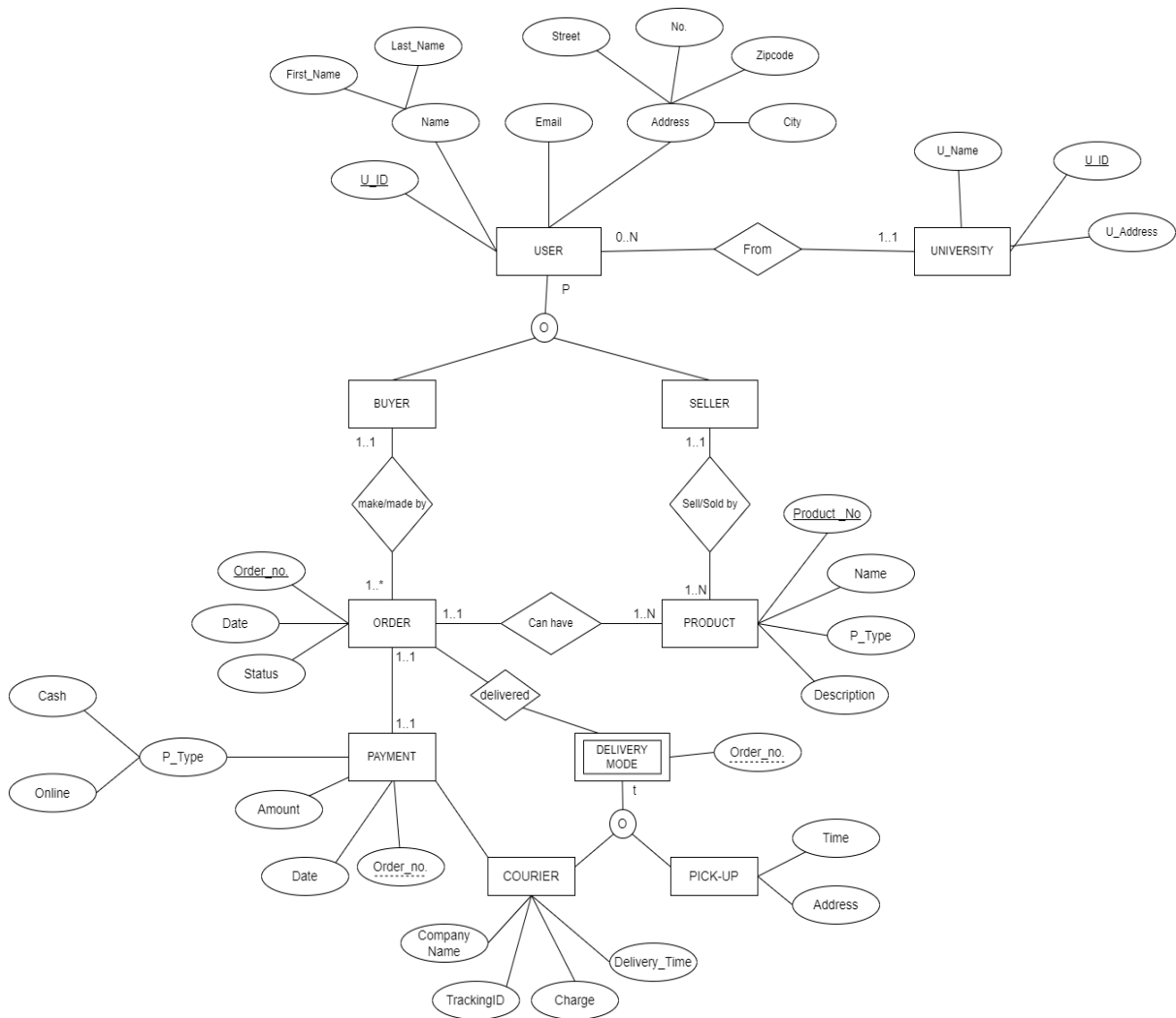
The platform will be open to all universities and will be designed to be easy to use and navigate for students. It will include features such as item listings, user profiles, messaging system, payment gateway and more, which will ensure a smooth transaction process. The platform will also be designed to be safe and secure, with measures put in place to protect both buyers and sellers.

Our project will have several benefits for the student community. Firstly, it will provide a reliable source for students to purchase household items at an affordable price. Secondly, it will allow students to earn money by selling items that they no longer need. Thirdly, it will help to reduce waste and promote sustainability by encouraging students to recycle and reuse household items. Lastly, it will create a sense of community among university students from different institutions as students will be able to interact with each other and build meaningful relationships.

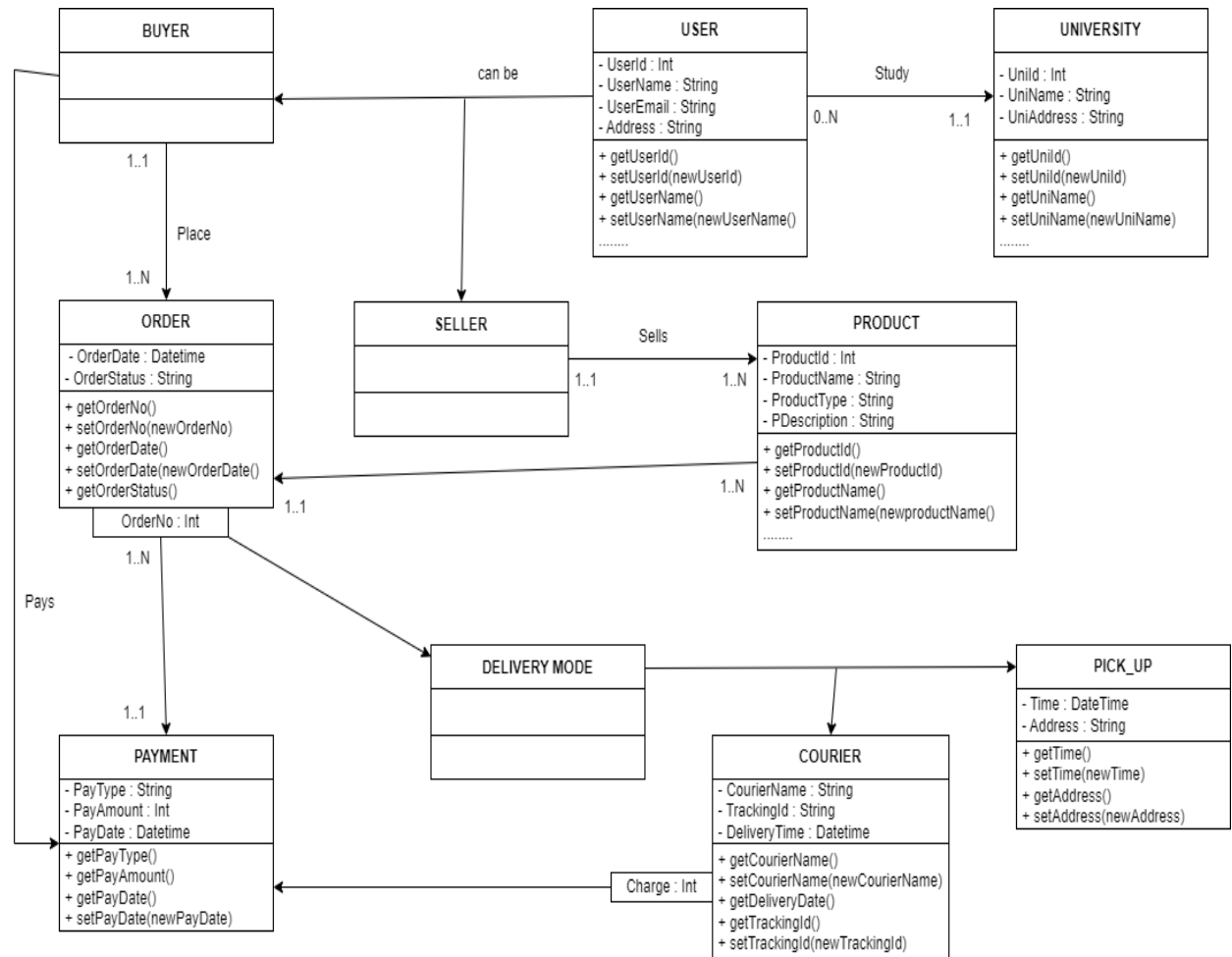
Overall, our project aims to make the process of buying and selling items easy, safe, and secure for the student community across all universities. We believe that this marketplace will provide a valuable service to students and will have a positive impact on their lives.

# CONCEPTUAL MODELING

## 1) EER DIAGRAM



## 2) UML DIAGRAM



## RELATIONAL MODEL

PrimaryKey – Bold , Foreign Key – with underscore

1. University (**UniID**, Uni\_Name, Uni\_Address)
2. User (**UserID**, FirstName, LastName, Email, Address)
3. Seller (**SellerID**, SUniID)
4. Buyer (**BuyerID**, BUniID)
5. Product (**ProdNo**, ProdName, PType, Description, SellerID, OrderID)
6. Order (**OrderID**, OrderDate, OrderStatus, PayID)
7. Payment (**PayID**, Amount, PayType)
8. Delivery (**DelID**, OrderNo)
9. Pick-Up (**PickupID**, POrderNo)
10. Courier (**CourierID**, Company, TrackingID, Charge, DelTime, OrderNo)

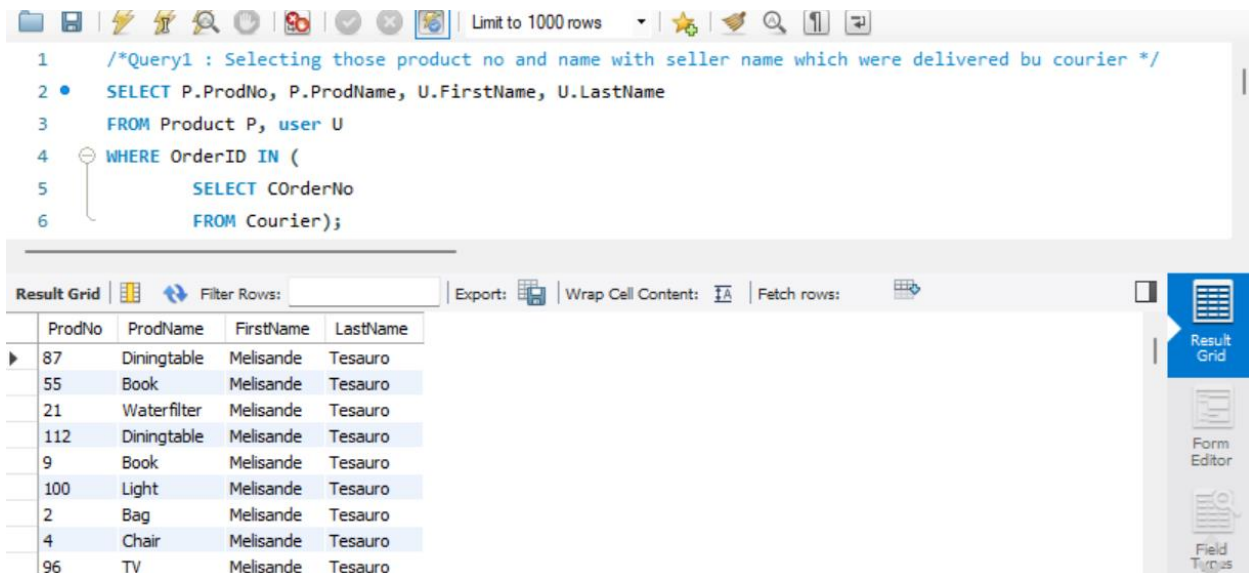
## IMPLEMENTATION IN MYSQL

Query 1 : Selecting those product no and name with seller name which were delivered by courier

```
SELECT P.ProdNo, P.ProdName, U.FirstName, U.LastName
```

```
FROM Product P, user U
```

```
WHERE OrderID IN ( SELECT COrderNo  
                   FROM Courier)
```



The screenshot displays a MySQL query editor with the following SQL query:

```
1 /*Query1 : Selecting those product no and name with seller name which were delivered bu courier */  
2 SELECT P.ProdNo, P.ProdName, U.FirstName, U.LastName  
3 FROM Product P, user U  
4 WHERE OrderID IN (  
5     SELECT COrderNo  
6     FROM Courier);
```

Below the query editor, the 'Result Grid' shows the following data:

	ProdNo	ProdName	FirstName	LastName
▶	87	Diningtable	Melisande	Tesauro
	55	Book	Melisande	Tesauro
	21	Waterfilter	Melisande	Tesauro
	112	Diningtable	Melisande	Tesauro
	9	Book	Melisande	Tesauro
	100	Light	Melisande	Tesauro
	2	Bag	Melisande	Tesauro
	4	Chair	Melisande	Tesauro
	96	TV	Melisande	Tesauro

Query 2 : User's name who are both buyer and seller

```
SELECT FirstName, LastName
```

```
FROM user
```

```
WHERE UserID in (SELECT SellerID FROM seller) AND  
               UserID in (SELECT BuyerID FROM buyer)
```

Limit to 1000 rows

```

7
8  /*Query2 : User's name who are both buyer and seller*/
9  • SELECT FirstName, LastName
10 FROM user
11 WHERE UserID in (SELECT SellerID FROM seller) AND UserID in (SELECT BuyerID FROM buyer);
12

```

Result Grid

FirstName	LastName
Bernie	Nolda
Alma	Huguet
Gayle	Caramuscia
Tami	Yemm
Sollie	Lorain
Abbe	Bette
Tierney	Airton
Tamarra	Torre
Titos	Hatherleigh

Result Grid  
Form Editor  
Field Trips

Query 3 : University which have most buyers

```

SELECT u.UniName, COUNT(b.BuyerID) AS No_Of_Buyers
FROM university u JOIN buyer b ON u.UniID = b.BUniID
GROUP BY u.UniName
ORDER BY No_Of_Buyers DESC
LIMIT 1;

```

Limit to 1000 rows

```

13  /*Query3 : University which have most buyers*/
14  • SELECT u.UniName, COUNT(b.BuyerID) AS No_Of_Buyers
15 FROM university u
16 JOIN buyer b ON u.UniID = b.BUniID
17 GROUP BY u.UniName
18 ORDER BY No_Of_Buyers DESC
19 LIMIT 1;

```

Result Grid

UniName	No_Of_Buyers
Universit�� Fran��saise d'��gypte	5

Result Grid  
Form Editor

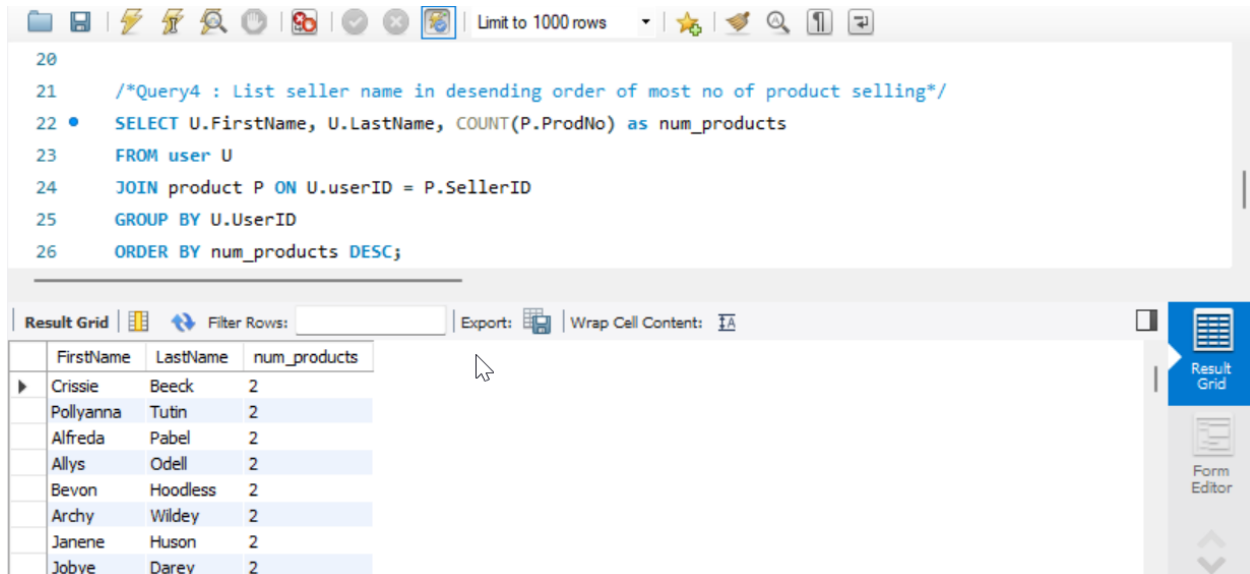
Query 4 : List seller name in descending order of most no of product selling

```
SELECT U.FirstName, U.LastName, COUNT(P.ProdNo) as num_products
```

```
FROM user U JOIN product P ON U.userID = P.SellerID
```

```
GROUP BY U.UserID
```

```
ORDER BY num_products DESC;
```



The screenshot shows a database query editor interface. The top toolbar includes icons for file operations, a search icon, and a dropdown menu set to "Limit to 1000 rows". The SQL editor contains the following query:

```
20  
21      /*Query4 : List seller name in descending order of most no of product selling*/  
22 •    SELECT U.FirstName, U.LastName, COUNT(P.ProdNo) as num_products  
23      FROM user U  
24      JOIN product P ON U.userID = P.SellerID  
25      GROUP BY U.UserID  
26      ORDER BY num_products DESC;
```

Below the editor is the "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The results are displayed in a table with the following data:

	FirstName	LastName	num_products
▶	Crissie	Beeck	2
	Pollyanna	Tutin	2
	Alfreda	Pabel	2
	Allys	Odell	2
	Bevon	Hoodless	2
	Archy	Willey	2
	Janene	Huson	2
	Jobye	Darey	2

On the right side of the interface, there are buttons for "Result Grid" and "Form Editor".

Query 5 : Number of products delivered by courier company

```
SELECT CompanyName, COUNT(*) AS Order_delivered
```

```
FROM courier
```

```
GROUP BY CompanyName
```

```
ORDER BY Order_delivered DESC
```



Limit to 1000 rows

```

28  /*Query5 : Number of products delivered by courier company*/
29  • SELECT CompanyName, COUNT(*) AS Order_delivered
30  FROM courier
31  GROUP BY CompanyName
32  ORDER BY Order_delivered DESC;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	CompanyName	Order_delivered
▶	USPS	12
	FedEx	8
	UPS	8

Result Grid  
Form Editor  
Field Types

Query 6 : Retrieves a record having highest delivery charge

SELECT \*

FROM courier

where charge >= ALL (SELECT charge

FROM courier);

Limit to 1000 rows

```

33
34  • SELECT *
35  FROM courier
36  where charge >= ALL (SELECT charge
37  FROM courier);
38
39

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	CourierID	CompanyName	TrackingID	Charge	DelTime	COrderNo
▶	27	USPS	d0d07fc0-4a13-4222-b91c-d7ae9e79f8da	\$9.92	5	59667-0075
*	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid  
Form Editor

Query 7 : Retrieves all records from user who are not registered as seller

SELECT \*

FROM user

WHERE NOT EXISTS( SELECT SellerID

FROM seller

WHERE user.UserID = seller.SellerID);

The screenshot shows a database query editor window titled "DMA\_Project\_Group16". The query is as follows:

```
38
39 • SELECT *
40 FROM user
41 WHERE NOT EXISTS( SELECT SellerID
42 FROM seller
43 WHERE user.UserID = seller.SellerID);
44
```

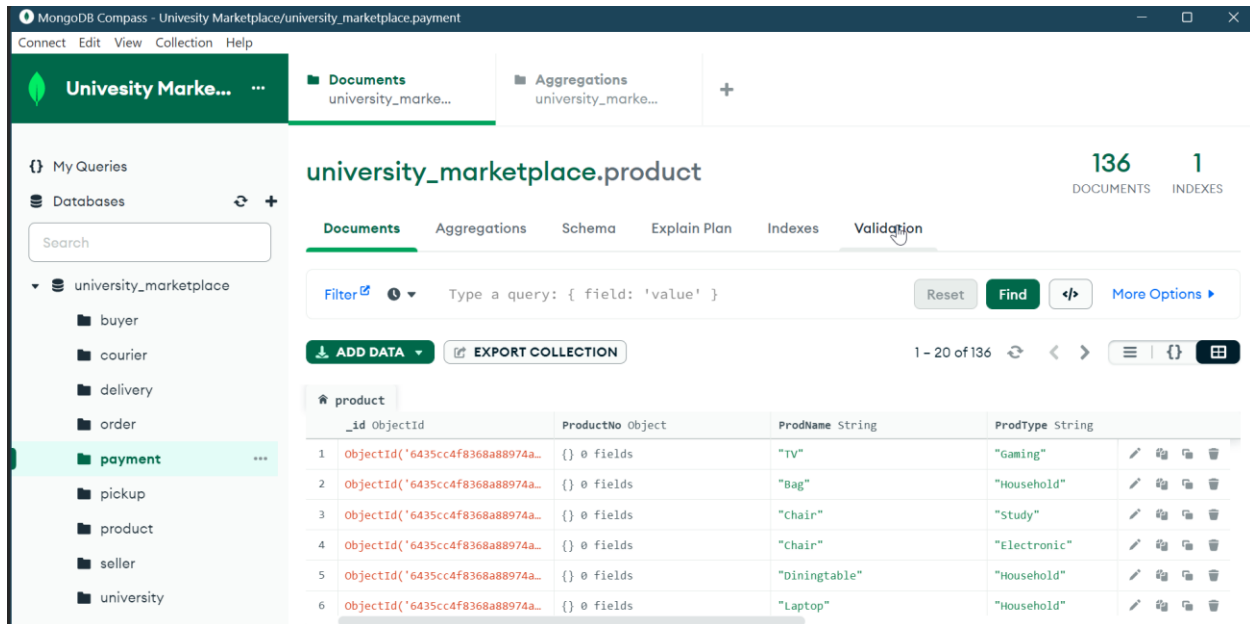
Below the query editor is a "Result Grid" showing the results of the query. The grid has five columns: UserID, FirstName, LastName, Email, and UserAddress. The results are as follows:

	UserID	FirstName	LastName	Email	UserAddress
▶	2401673	Iona	Napthine	inapthinee@patch.com	950 Waywood Avenue
	19867980	Lazarus	Shallow	lshallowlw@list-manage.com	82747 Huxley Alley
	71835741	Harland	Nerval	hnervaldp@tuttocitta.it	3 Charing Cross Place
	86529773	Annemarie	Siggin	asiggin88@cargocollective.com	3 Grayhawk Park
	100560954	Beatrice	Jenicke	bjenicke2u@bloglovin.com	1701 Sherman Lane
	104127147	Emiline	Niave	eniaveeg@php.net	48 Prairieview Lane
	106271377	Quinta	Czajkowski	qczajkowskipl@joomla.org	032 Del Mar Street

At the bottom of the window, there is a status bar showing "Result 212", "user 213", "Result 214", "Result 215", "Result 216", "courier 217", "user 218", and an "Apply" button.

# IMPLEMENTATION IN NOSQL

## Preview of project dataset into MongoDB



**Query1 :** This query gives total purchase value by payment type category i.e. how much amount of product purchased through cash-on-delivery and how much by prepaid.

```
{
  _id: "$PayType",
  Amount: {
    $sum: "$Amount($)"
  }
}
```

**university\_marketplace.payment** 250 DOCUMENTS 1 INDEXES

Documents **Aggregations** Schema Explain Plan Indexes Validation

Pipeline **\$group** Explain Export Run More Options ▶

query1 **SAVE** **CREATE NEW** **EXPORT TO LANGUAGE** **PREVIEW** **STAGES** **TEXT**

```

1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   _id: "$PayType",
7   Amount: {
8     $sum: "$Amount($)",
9   },
10 }

```

Output after **\$group** stage (Sample of 2 documents)

_id	Amount
"Cash-on-delivery"	34007.59

**Query2** : This query will return all the records having product type “Study” and product name “Chair” from the product table.

```

{
  $and: [
    { ProdType: "Study"},
    { ProdName: "Chair"}
  ]
}

```

**university\_marketplace.product** 136 DOCUMENTS 1 INDEXES

Documents **Aggregations** Schema Explain Plan Indexes Validation

Pipeline **\$match** Explain Export Run More Options ▶

Untitled - modified **SAVE** **CREATE NEW** **EXPORT TO LANGUAGE** **PREVIEW** **STAGES** **TEXT**

```

1 /**
2  * query: The query in MQL.
3  */
4 {
5   $and : [
6     {'ProdType' : 'Study'},
7     {'ProdName' : 'Chair'}
8   ]
9 }

```

Output after **\$match** stage (Sample of 3 documents)

_id	ProductNo	ProdName	ProdType	ProdDescription	SellerID
ObjectId('6435cc4f8368a88974a9bdf')	Object	"Chair"	"Study"	"xyz"	1527537382

### Query3 :

Stage1 : this query will return all the records which took 7 or more days to delivered

```
{  
  DelTime: {  
    $gte: 7}  
}
```

Stage2 : This query will work on output of stage1 and provide company wise total no of delayed products

```
{_id: "$CompanyName",  
  Deliveries: {  
    $count: { }  
  }  
}
```

The screenshot displays the MongoDB Atlas web interface for the 'university\_marketplace.courier' database. The left sidebar shows a list of databases, with 'courier' selected. The main panel shows the 'Aggregations' tab, where a pipeline is being edited. The pipeline consists of a single stage, '\$match', with the following query:

```
1 /**  
2  * query: The query in MQL.  
3  */  
4 {  
5   'DelTime': { $gte: 7 },  
6 }  
7
```

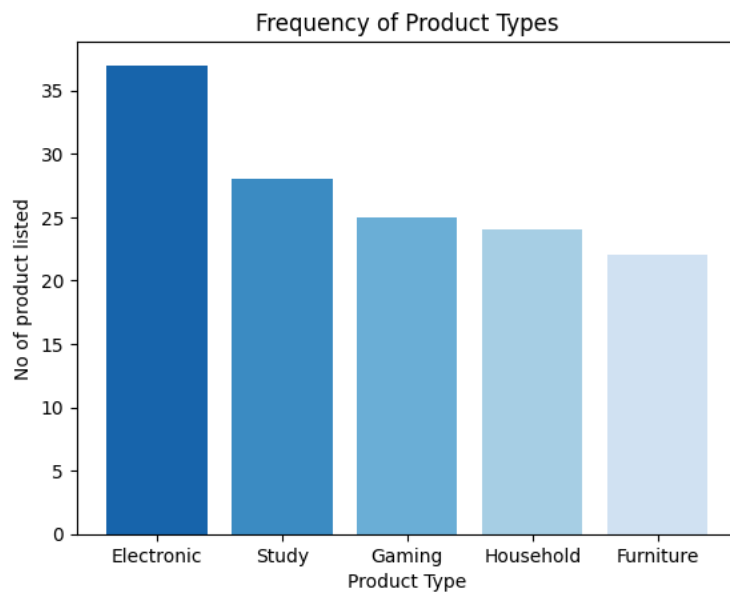
The interface also shows the 'Query3' name, a 'SAVE' button, a '+ CREATE NEW' button, and an 'EXPORT TO LANGUAGE' button. The 'PREVIEW' button is active, showing a sample of 10 documents after the '\$match' stage:

```
{  
  "_id": "ObjectId('6435cbfb8368a88974a9bc51')",  
  "CourierID": 56,  
  "CompanyName": "USPS",  
  "TrackingID": "56c85dd1-ed00-441b-b465-616895eea2dc",  
  "Charge": "$5.70",  
  "DelTime": 8,  
  "OrderNo": "57344-160"  
}
```

## DATABASE ACCESS VIA PYTHON

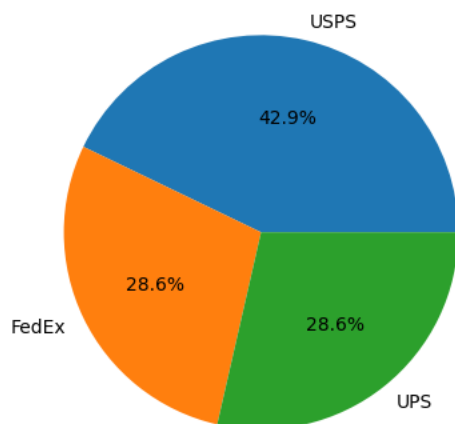
To access the database, Python and the pymysql.connect library are used. The pd.read\_sql\_query method is used to run and retrieve the results of a query, which are already converted into a dataframe. Finally, matplotlib is used to visualize the analyzed data.

Below bar graph represents category wise number of products listed in our database.



Below pie chart represents percentage of product delivered by all courier companies.

Proportion of Products Delivered by Company



Below table represents seller name who sold most number of products

	FirstName	LastName	num_products
0	Crissie	Beeck	2
1	Pollyanna	Tutin	2
2	Alfreda	Pabel	2
3	Allys	Odell	2
4	Bevon	Hoodless	2

## FUTURE IMPLEMENTATION

We plan to add features to our online student marketplace, including image uploading for sellers, a messaging system for communication, and a bidding system for buyers. These additions will improve functionality and user experience. Sellers will be able to upload multiple images to showcase their items, while a messaging system will allow secure communication between buyers and sellers. The bidding system will provide a fair and exciting way for buyers to potentially get items at a lower price. These features will enhance our platform, making it easier and safer for students to buy and sell items.