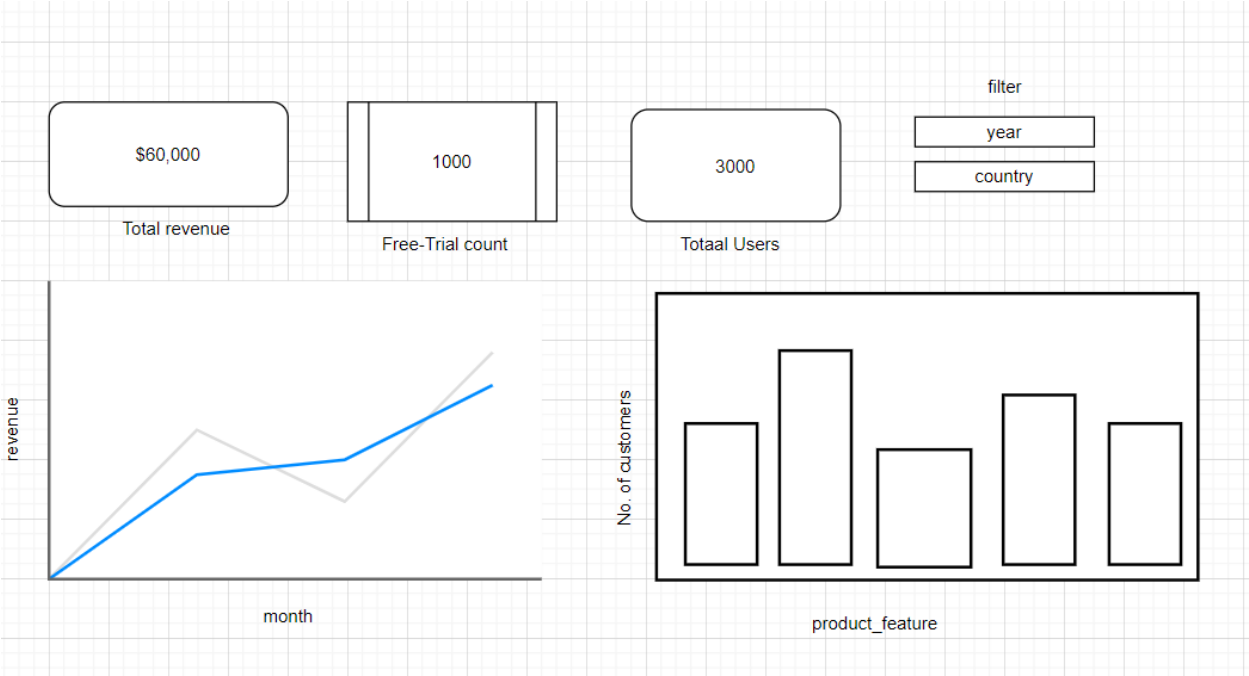
1. A high-level KPI dashboard that shows revenue, the number of trials we have in the pipeline, and how many customers are using the most important product features.




```
WITH features AS (
  SELECT
    created_by_id,
    CASE WHEN drawing_name IS NOT NULL THEN 'Drawing' ELSE 'Document' END AS
feature_type
  FROM
    dim_drawings
    FULL OUTER JOIN dim_documents ON dim_drawings.created_by_id =
dim_documents.created_by_id
)
SELECT
  features.feature_type AS feature_name,
  SUM(CASE WHEN is_trial = 0 THEN 1 ELSE 0 END) AS trial_count
  COUNT(DISTINCT dim_users.user_id) AS user_count,
  COUNT(DISTINCT dim_customers.customer_id) AS customer_count,
  DATE_TRUNC('month', subscriptions_start) AS month,
  SUM(subscriptions_amount) AS revenue
FROM
  features
  FULL OUTER JOIN dim_users ON feature_usage.created_by_id = dim_users.user_id
```

```
    FULL OUTER JOIN dim_customers ON dim_users.customer_id =
dim_customers.customer_id
    FULL OUTER JOIN dim_subscriptions ON dim_customers.subscription_id =
dim_subscriptions.subscription_id
GROUP BY
    features.feature_type,
    DATE_TRUNC('month', subscriptions_start);
```

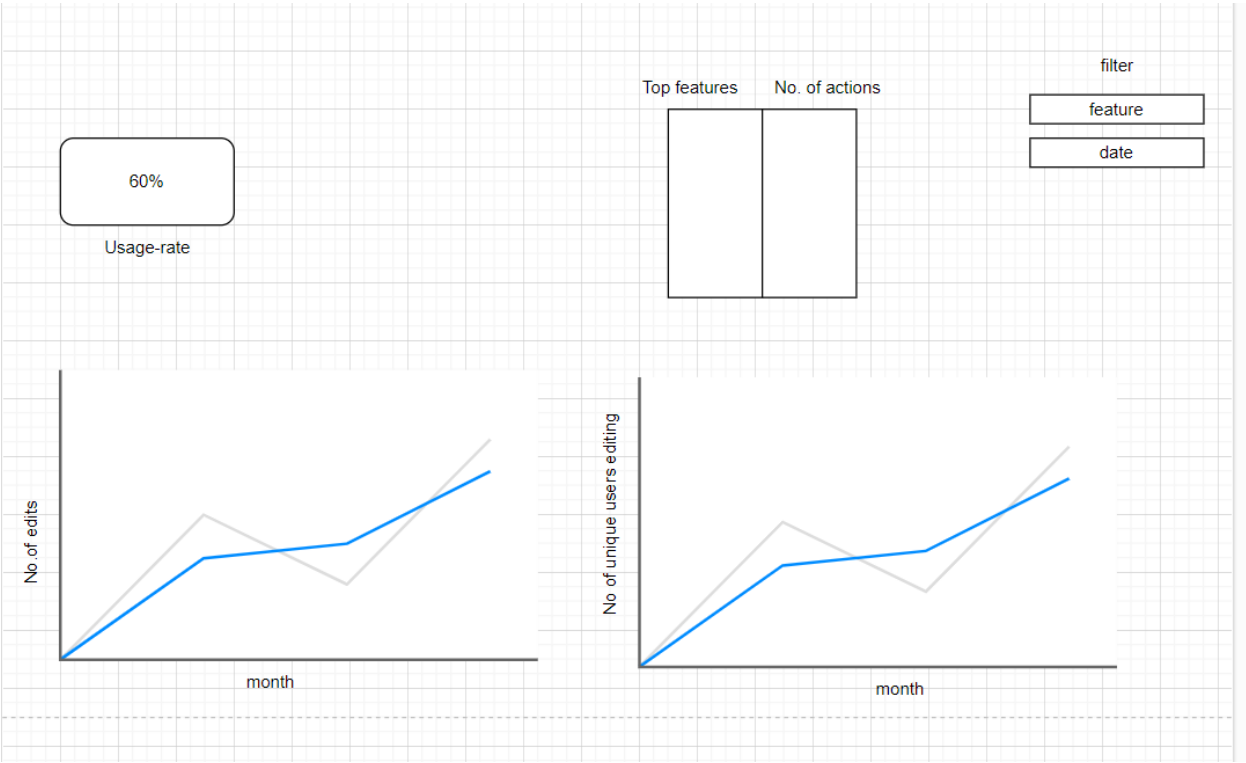
Rpt_Kpi		
Definition	This table stores information about revenue, trials and product feature usage.	
Grain	One row per feature type per month	
Key Columns	month: The month of feature usage or subscription. feature: Unique features of the product. revenue: The total dollar amount of subscription	
Source Tables	Dim_users,fact_drawings,dim_subscriptions,dim_customers	
Additivity	The revenue column is additive per unique month.	
Suggested Usage	This table can be used to analyze revenue, trial count and feature usage.	

Sample table output

feature_name	trial_count	user_count	customer_count	month	revenue
Document	5	10	10	2022-01-01	300.0
Drawing A	5	20	15	2022-01-01	250.0

Null	1	2	1	2022-01-01	60.0
Document A	10	6	20	2022-02-01	45.0
Drawing B	10	4	2	2022-02-01	25.0
Null	2	1	6	2022-02-01	400.0

2. A low-level product analytics dashboard so that our Product Team can gauge the health of individual features. Are our customers using the product how we think they should?



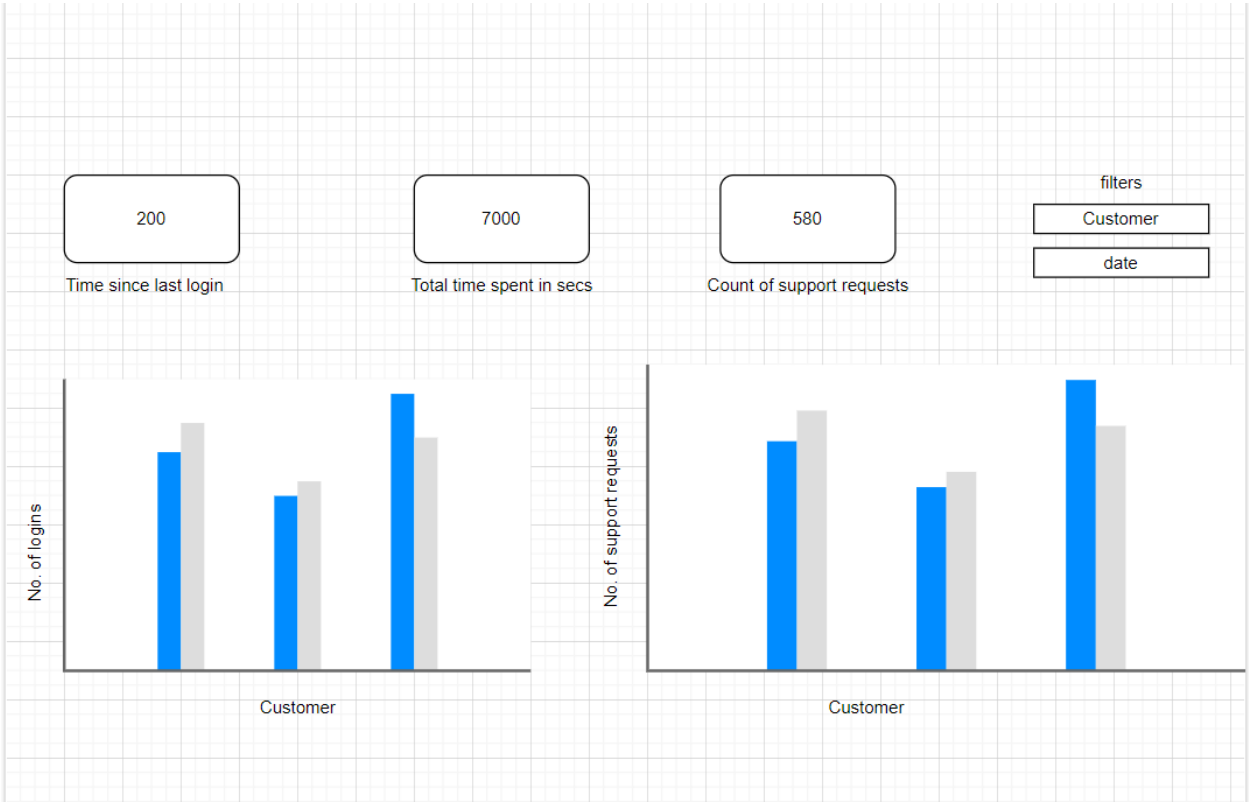
```
WITH features AS (
  SELECT
    created_by_id,
    CASE WHEN drawing_name IS NOT NULL THEN 'Drawing' ELSE 'Document' END AS
feature_type
  FROM
    dim_drawings
    FULL OUTER JOIN dim_documents ON dim_drawings.created_by_id =
dim_documents.created_by_id
),
drawing_edits AS (
  SELECT
    created_by_id,
    COUNT(*) AS num_edits
  FROM
    fact_drawing_edits
  GROUP BY
    created_by_id
),
document_edits AS (
  SELECT
    created_by_id,
    COUNT(*) AS num_edits
  FROM
    fact_document_edits
  GROUP BY
    created_by_id
)
SELECT
  features.feature_type AS feature_name,
  SUM(CASE WHEN is_trial = 0 THEN 1 ELSE 0 END) AS trial_count,
  COUNT(DISTINCT dim_users.user_id) AS user_count,
  DATE_TRUNC('month', COALESCE(subscriptions_start, CURRENT_DATE)) AS month,
  COALESCE(SUM(drawing_edits.num_edits), 0) AS drawing_edit_count,
  CASE WHEN features.feature_type = 'Document' THEN
COALESCE(SUM(document_edits.num_edits), 0) ELSE NULL END AS document_edit_count,
  COUNT(DISTINCT CASE WHEN drawing_edits.num_edits > 0 THEN
drawing_edits.created_by_id END)
```

```
    + COUNT(DISTINCT CASE WHEN document_edits.num_edits > 0 THEN
document_edits.created_by_id END)
    AS unique_editing_users_count
FROM
  feature_usage
  FULL OUTER JOIN dim_users ON features.created_by_id = dim_users.user_id
  FULL OUTER JOIN dim_customers ON dim_users.customer_id = dim_customers.customer_id
  FULL OUTER JOIN dim_subscriptions ON dim_customers.subscription_id =
dim_subscriptions.subscription_id
  LEFT JOIN drawing_edits ON features.created_by_id = drawing_edits.created_by_id
  LEFT JOIN document_edits ON feature_usage.created_by_id = document_edits.created_by_id
GROUP BY
  feature_usage.feature_type,
  DATE_TRUNC('month', COALESCE(subscriptions_start, CURRENT_DATE))
ORDER BY
  drawing_edit_count + COALESCE(document_edit_count, 0) DESC;
```

Rpt_Feature_Usage		
Definition	This table stores information about usage of the Drawing and Document features in VeggieMart's application.	
Grain	One row per feature type per month	
Key Columns	store_id: A unique identifier for each store. region: The region where the store is located. manager_id: The unique identifier of the store's manager. store_name: The name of the store.	
Primary Key	feature_name month	
Source Tables		
Additivity		
Suggested Usage	This table can be used to analyze the usage and performance of the Drawing and Document features over time, and to identify areas for improvement or further development.	

feature_name	trial_count	user_count	month	drawing_edit_count	document_edit_count	unique_editing_users_count
Drawing A	5	23	9/1/2022	100	0	20
Document A	14	17	9/1/2022	0	80	15
Drawing D	10	27	8/1/2022	120	0	25
Document C	9	19	8/1/2022	0	70	12
Drawing B	8	30	7/1/2022	150	0	30
Document E	11	20	7/1/2022	0	90	18

3. I added two fact tables to the model design, The fact_user_logins, fact_support_requests.




```
SELECT
    dim_users.user_id,
    dim_customers.customer_id,
```

```
    DATE_TRUNC('month', fact_user_logins.login_time) AS month,
    COUNT(*) AS num_logins,
    SUM(fact_user_logins.user_session_duration) AS total_session_duration,
    COUNT(fact_support_requests.support_request_id) AS num_support_requests
FROM
    fact_user_logins
    JOIN dim_users ON fact_user_logins.user_id = dim_users.user_id
    JOIN dim_customers ON dim_users.customer_id = dim_customers.customer_id
    LEFT JOIN fact_support_requests ON dim_users.user_id = fact_support_requests.user_id
GROUP BY
    dim_users.user_id,
    dim_customers.customer_id,
    DATE_TRUNC('month', fact_user_logins.login_time)
ORDER BY
    dim_customers.customer_id,
    dim_users.user_id,
    DATE_TRUNC('month', fact_user_logins.login_time)
```

Rpt_Customer_Activity	
Definition	This table stores information about customer activity, including login events and support requests.
Grain	One row per customer per month.
Key Columns	customer_id: A unique identifier for each customer. month: The month in which the activity occurred.
Primary Key	customer_id, month
Source Tables	dim_customers , dim_users ,fact_support_requests,fact_user_logins
Additivity	support_requests and user_session_durations columns are both additive
Suggested Usage	This table can be used to analyze customer activity patterns and identify potential churn risks. It can also be used to track the effectiveness of support services and user engagement over time.

month	customer_name	user_name	num_logins	total_duration_secs	num_support_requests
2022-07-01	ACME Corp	David Hundeyin	10	3600	30
2022-07-01	APPLE Corp	Ifunanya Orji	8	2400	1
2022-08-01	DOJO Corp	Aimanehi Oge	12	4800	2
2022-08-01	ACME Corp	Jane Chinaka	15	5400	70
2022-10-01	ACME Corp	John Ifeanyi	5	1800	2
2023-01-01	DOMO Corp	Michael Amara	10	3600	1
2023-01-01	PEPSI Corp	David Tope	8	3000	0