

Database Project

M#1

System Requirements

User

- **UserID:** Unique identifier for each user to ensure distinct records in the database.
 - **First Name and Last Name:** User's personal details for identification.
 - **Profile Name:** A unique name chosen by the user to represent their account.
 - **Bio:** A brief description about the user, visible on their profile.
 - **Sign-Up Date:** Timestamp recording when the user created their account.
-

Followers

- **FollowingUserID:** Foreign key linking the user who is following another user.
 - **FollowedUserID:** Foreign key linking the user who is being followed.
 - **Created At:** Timestamp recording when the follower relationship was established.
 - **Status:** Tracks the status of the follower relationship (e.g., active or blocked).
-

Chats

- **ChatID:** Unique identifier for each chat to ensure distinct records.
 - **Chat Name:** Optional name assigned to the chat (e.g., group name).
 - **Created At:** Timestamp recording when the chat was created.
 - **Role:** Defines user roles within the chat, such as admin or member.
-

Chat Participants

- **ChatID:** Foreign key linking to the associated chat.
- **UserID:** Foreign key linking to the user in the chat.
- **Joined At:** Timestamp recording when the user joined the chat.

- **Role:** Defines the user's role in the chat (e.g., admin, member).
-

Messages

- **MessageID:** Unique identifier for each message to ensure distinct records.
 - **SenderID:** Foreign key linking to the user who sent the message.
 - **ReceiverID:** Foreign key linking to the user or chat receiving the message.
 - **Content:** The text or content of the message.
 - **Timestamp:** Records the exact time the message was sent.
 - **Read Status:** Tracks whether the message has been read.
-

Attachments

- **AttachmentID:** Unique identifier for each attachment.
 - **MessageID:** Foreign key linking to the associated message.
 - **File Path:** Path to the file stored in the system.
 - **File Type:** Indicates the type of file (e.g., image, video, document).
 - **Uploaded At:** Timestamp recording when the file was uploaded.
-

Posts

- **PostID:** Unique identifier for each post to ensure distinct records.
 - **Created By UserID:** Foreign key linking the post to its creator.
 - **Created DateTime:** Timestamp recording when the post was created.
 - **Caption:** Text accompanying the post.
 - **Post Type:** Type of post, such as image, video, or text.
-

Post Media

- **PostMediaID:** Unique identifier for each media file in a post.
- **PostID:** Foreign key linking the media to the associated post.
- **FilterID:** Foreign key linking to the applied filter.
- **Media File:** File path of the media file.
- **Position:** Order or placement of the media within the post.

Post Media User Tag

- **PostMediaID:** Foreign key linking to the associated post media.
 - **UserID:** Foreign key linking to the tagged user.
 - **X Coordinate:** Horizontal position of the tag on the media.
 - **Y Coordinate:** Vertical position of the tag on the media.
-

Reactions

- **UserID:** Foreign key linking to the user who reacted.
 - **PostID:** Foreign key linking to the associated post.
-

Comments

- **CommentID:** Unique identifier for each comment.
 - **PostID:** Foreign key linking the comment to the associated post.
 - **Created By UserID:** Foreign key linking the comment to the user who created it.
 - **Created DateTime:** Timestamp recording when the comment was created.
 - **Comment Text:** The actual content of the comment.
 - **Comment Replied To ID:** Foreign key linking to the parent comment for replies.
-

Effects

- **EffectID:** Unique identifier for each effect.
 - **Effect Name:** Name of the effect (e.g., "Vintage", "Black and White").
-

Post Effect

- **PostMediaID:** Foreign key linking to the associated post media.
 - **EffectID:** Foreign key linking to the applied effect.
 - **Scale:** Value representing the intensity or scale of the effect.
-

Filters

- **FilterID:** Unique identifier for each filter.
- **Filter Name:** Name of the filter (e.g., "Sepia", "Blur").
- **Filter Details:** Descriptive details about the filter settings or effects.

Non-Functional Requirements:

1. Performance:

- Ensure efficient querying of relational data between entities like posts, tags, and users for real-time updates (improves responsiveness and user experience).
- Optimize data indexing strategies for frequently queried fields like UserID and PostID (reduces query execution time).

2. Scalability:

- Incorporate horizontal scaling solutions for handling relationships like followers and chats (supports growing user base).
- Use sharding techniques for large datasets to distribute storage and computation loads (enhances scalability).

3. Security:

- Restrict access to sensitive entities (e.g., BusinessAccountsContactInfo) to authenticated users (ensures data protection and privacy).
- Encrypt user passwords and sensitive data at rest and in transit (strengthens security measures).

4. Usability:

- Provide detailed analytics dashboards for businesses to monitor products, ads, and engagement (enhances user insights and satisfaction).
- Include tooltips and inline help for complex features (improves user onboarding and understanding).

5. Data Integrity:

- Ensure referential integrity between related tables, such as UserID in the Followers, Posts, and Comments tables (maintains data consistency).

- Validate data inputs at both client-side and server-side to prevent invalid or malicious entries (improves system reliability).

6. Availability:

- Ensure high availability for critical services like notifications, messaging, and sentiment analysis (minimizes downtime).
- Deploy redundant systems and failover mechanisms to prevent service interruptions (enhances reliability).

7. Latency Optimization:

- Optimize system responses to minimize latency in user interactions, particularly in high-demand features like messaging and notifications (enhances user satisfaction).
- Utilize content delivery networks (CDNs) for faster content loading, especially for media files (improves user experience).

8. Disaster Recovery:

- Implement robust backup and recovery systems to ensure minimal data loss in case of hardware or software failure (ensures reliability).
- Schedule regular testing of disaster recovery processes to ensure readiness (improves system preparedness).

9. Compatibility:

- Ensure compatibility across a wide range of devices and browsers, including mobile and desktop platforms (maximizes accessibility).
- Support legacy browsers to accommodate older devices while encouraging modern standards (extends user reach).

10. Accessibility Compliance:

- Design interfaces to be accessible for users with disabilities, meeting WCAG (Web Content Accessibility Guidelines) standards (ensures inclusivity).
- Provide screen reader support and high-contrast modes for visually impaired users (enhances accessibility).

11. Logging and Monitoring:

- Implement comprehensive logging and monitoring tools to track system health, errors, and user activity (supports maintenance and troubleshooting).

- Generate automated alerts for unusual patterns or performance issues (ensures proactive issue resolution).

12. Energy Efficiency:

- Optimize resource usage in data centers to align with sustainable energy practices (promotes environmental responsibility).
- Monitor and report energy consumption metrics to identify areas for improvement (enhances operational sustainability).

Stakeholders Requirements

Primary Stakeholders

1. Users:

- Have several information, unique id, first and last name, profile shown name, bio, and his signing up date
- Individual users who create accounts, interact with others, and participate in chats.
- Content creators and influencers who use the platform to reach followers.
- Users engaging with posts through reactions, comments, and media tagging.

2. Business Owners/Management:

- Decision-makers responsible for the overall strategy and growth of the platform.

3. Developers/Engineering Team:

- Responsible for building, maintaining, and scaling the platform's architecture, including post media and tagging features.

4. UI/UX Designers:

- Ensure the platform is visually appealing and easy to navigate, especially for post creation, tagging, and media effects.

5. Database Administrators:

- Manage database optimization, backups, and performance tuning for post, media, and interaction data.

6. Quality Assurance (QA) Team:

- Test the system for bugs, security vulnerabilities, and performance issues across functionalities like tagging and reactions.

Secondary Stakeholders

1. Advertisers:

- Utilize posts and media features to create engaging campaigns with tagged users and media effects.

2. Regulatory Authorities:

- Ensure the platform complies with laws governing user-generated content, reactions, and tagging.

3. Investors:

- Stakeholders funding the project and expecting returns based on performance and user engagement.

4. Support and Customer Service Team:

- Assist users with issues related to posts, tagging, filters, and comments.

5. Third-party Service Providers:

- Providers of APIs or tools for media storage, content delivery, and analytics.

M#2

- The ERD

M#3

- The DB schema
- The Data dictionary

TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	IS_NULLABLE
ad_insight	ad_id	int	None	NO
ad_insight	views_count	int	None	YES
ad_insight	clicks_count	int	None	YES
ad_insight	impressions	int	None	YES
ad_insight	engagement_rate	decimal	None	YES
ad_insight	cost_per_click	decimal	None	YES
ad_insight	cost_per_mille	decimal	None	YES
advertisement	ad_id	int	None	NO
advertisement	created_by_user_id	int	None	YES
advertisement	post_id	int	None	YES
advertisement	product_id	int	None	YES
advertisement	created_datetime	datetime	None	YES
advertisement	target_audience	text	2147483647	YES
advertisement	start_date	date	None	YES
advertisement	end_date	date	None	YES
attachment	id	int	None	NO
attachment	message_id	int	None	YES
attachment	file_path	varchar	255	NO
attachment	file_type	varchar	50	YES
attachment	uploaded_at	datetime	None	YES
business_account	business_id	int	None	NO

business_account	business_name	varchar	100	NO
business_account	category	varchar	100	YES
business_account	website	varchar	255	YES
business_account	contact_email	varchar	100	YES
business_account	contact_phone	varchar	15	YES
chat_participants	chat_id	int	None	NO
chat_participants	user_id	int	None	NO
chat_participants	joined_at	datetime	None	YES
chat_participants	role	varchar	50	YES
chats	chat_id	int	None	NO
chats	chat_name	varchar	100	YES
chats	created_at	datetime	None	YES
collection	collection_id	int	None	NO
collection	user_id	int	None	YES
collection	collection_name	varchar	100	YES
collection	created_at	datetime	None	YES
collection_post	collection_id	int	None	NO
collection_post	post_id	int	None	NO
comment	id	int	None	NO
comment	created_by_user_id	int	None	YES
comment	post_id	int	None	YES
comment	created_datetime	datetime	None	YES

comment	comment	text	2147483647	NO
comment	comment_replied_to_id	int	None	YES
effect	id	int	None	NO
effect	effect_name	varchar	100	YES
filter	id	int	None	NO
filter	filter_name	varchar	100	YES
filter	filter_details	text	2147483647	YES
follower	following_user_id	int	None	NO
follower	followed_user_id	int	None	NO
follower	created_at	datetime	None	YES
follower	status	varchar	50	YES
message	message_id	int	None	NO
message	sender_id	int	None	YES
message	receiver_id	int	None	YES
message	content	text	2147483647	NO
message	time_stamp	datetime	None	YES
message	is_read	bit	None	YES
post	post_id	int	None	NO
post	created_by_user_id	int	None	YES
post	created_datetime	datetime	None	YES
post	caption	text	2147483647	YES
post	location	varchar	255	YES

post	post_type	int	None	YES
post_media	post_media_id	int	None	NO
post_media	post_id	int	None	YES
post_media	filter_id	int	None	YES
post_media	media_file	varchar	255	NO
post_media	longitude	int	None	YES
post_media	latitude	int	None	YES
post_media_effect	post_media_id	int	None	NO
post_media_effect	effect_id	int	None	NO
post_media_user_tag	post_media_id	int	None	NO
post_media_user_tag	user_id	int	None	NO
post_media_user_tag	x_coordinate	int	None	YES
post_media_user_tag	y_coordinate	int	None	YES
post_product_tag	post_id	int	None	NO
post_product_tag	product_id	int	None	NO
post_type	id	int	None	NO
post_type	post_type_name	varchar	50	YES
product	product_id	int	None	NO
product	business_id	int	None	YES
product	product_name	varchar	100	YES
product	price	decimal	None	YES
product	description	text	2147483647	YES

product	available_stock	int	None	YES
reaction	user_id	int	None	NO
reaction	post_id	int	None	NO
saved_post	user_id	int	None	NO
saved_post	post_id	int	None	NO
saved_post	saved_at	datetime	None	YES
user	user_id	int	None	NO
user	fname	varchar	100	NO
user	lname	varchar	100	YES
user	profile_name	varchar	100	NO
user	profile_pic_path	varchar	255	YES
user	email	varchar	100	NO
user	password	varchar	255	NO
user	bio	text	2147483647	YES
user	signup_date	datetime	None	YES
user	account_type	varchar	20	NO
user_activity	user_id	int	None	NO
user_activity	activity_type	varchar	50	YES
user_activity	related_post_id	int	None	NO
user_activity	created_at	datetime	None	YES

M#4

- SQL and Relational Algebra Reports

1. Users with the most followers

```
SELECT u.profile_name, COUNT(f.following_user_id) as follower_count
FROM user u
LEFT JOIN follower f ON u.user_id = f.followed_user_id
GROUP BY u.profile_name
HAVING COUNT(f.following_user_id) > 0
ORDER BY follower_count DESC;
```

Relational Algebra:

$\sigma_{\text{follower_count} > 0} (\pi_{\text{profile_name}, \text{follower_count}} (\gamma_{\text{profile_name}};$
 $\text{COUNT}(\text{following_user_id}) \rightarrow \text{follower_count} (\text{user} \bowtie [\text{user_id} = \text{followed_user_id}] \text{follower})))$

Output:

profile_name	follower_count	
ynolan	4	
waltonstephanie	4	
frankstafford	4	
xhebert	3	
andrew76	3	
brian18	3	
thomasdavis	3	
smithstephen	3	

2. Business accounts with highest product engagement in ads

```
SELECT ba.business_name,  
       COUNT(DISTINCT ad.ad_id) as total_ads,  
       SUM(ai.clicks_count) as total_clicks,  
       AVG(ai.engagement_rate) as avg_engagement  
FROM business_account ba  
JOIN product p ON ba.business_id = p.business_id  
JOIN advertisement ad ON p.product_id = ad.product_id  
JOIN ad_insight ai ON ad.ad_id = ai.ad_id  
GROUP BY ba.business_name  
ORDER BY avg_engagement DESC;
```

Relational Algebra:

π business_name, total_ads, total_clicks, avg_engagement (γ business_name;
COUNT(DISTINCT ad_id) \rightarrow total_ads, SUM(clicks_count) \rightarrow total_clicks,
AVG(engagement_rate) \rightarrow avg_engagement (business_account \bowtie [business_account.business_id =
product.business_id] product \bowtie [product.product_id = advertisement.product_id] advertisement
 \bowtie [advertisement.ad_id = ad_insight.ad_id] ad_insight)))

Output:

business_name	total_ads	total_clicks	avg_engagement
Ortega-Johnson	1	340	9.770000
Riley-Anderson	1	709	9.520000
Ward, Cannon and Banks	1	224	9.270000
Stone Ltd	1	1590	9.150000
Harvey-Snyder	1	1326	8.780000
Rivera Ltd	1	4449	8.230000
Wilson, Simpson and Li	1	240	8.120000
Woods, Snyder and Day	2	7910	7.615000
Young Group	1	2996	7.410000

3. Most engaging posts (combining likes and comments)

```
SELECT p.post_id, p.caption,
       COUNT(DISTINCT r.user_id) as likes,
       COUNT(DISTINCT c.id) as comments,
       (COUNT(DISTINCT r.user_id) + COUNT(DISTINCT c.id)) as total_engagement
FROM post p
LEFT JOIN reaction r ON p.post_id = r.post_id
LEFT JOIN comment c ON p.post_id = c.post_id
GROUP BY p.post_id, p.caption
ORDER BY total_engagement DESC;
```

Relational Algebra:

$\pi_{\text{post_id}, \text{caption}, \text{likes}, \text{comments}, (\text{likes} + \text{comments}) \rightarrow \text{total_engagement}} (\gamma_{\text{post_id}, \text{caption};$
 $\text{COUNT}(\text{DISTINCT user_id}) \rightarrow \text{likes}, \text{COUNT}(\text{DISTINCT id}) \rightarrow \text{comments} (\text{post} \bowtie [\text{post.post_id} =$
 $\text{reaction.post_id}] \text{reaction} \bowtie [\text{post.post_id} = \text{comment.post_id}] \text{comment}))$

Output:

post_id	caption	likes	comments	total_engagement
515	Serious forward enter once you consider nearly...	2	0	2
532	Specific myself break nor also factor north blue...	2	0	2
74	Shake itself live himself myself decision.	1	1	2
264	Yet around mouth southern recognize list yes w...	1	1	2
5	Player second although month program leg stag...	1	0	1
28	Agreement finish season end walk better set be...	1	0	1
30	Box support hot age mouth region fill.	1	0	1
45	Artist base down human three ten should attack...	1	0	1
46	Room front goal yeah maybe let table nature act...	1	0	1
49	Ahead involve care he sense price light.	1	0	1
112	Among writer together from recognize name after.	1	0	1

4. Rank products by sales stock for each business

```
SELECT
    business_account.business_name,
    product.product_name,
    product.available_stock,
    RANK() OVER (PARTITION BY product.business_id ORDER BY product.available_stock DESC) AS stock_rank
FROM product
JOIN business_account ON product.business_id = business_account.business_id;
```

Relational Algebra:

π business_name, product_name, available_stock, stock_rank (ρ [RANK() \rightarrow stock_rank] (RANK() OVER (PARTITION BY business_id ORDER BY available_stock DESC) (product \bowtie [product.business_id = business_account.business_id] business_account)))

Output:

business_name	product_name	available_stock	stock_rank
Hill-Gregory	See	80	1
Hill-Gregory	Rate	78	2
Hill-Gregory	Southern	75	3
Hill-Gregory	Recent	72	4
Hill-Gregory	Personal	70	5
Hill-Gregory	Music	54	6
Hill-Gregory	Society	52	7
Hill-Gregory	Thousand	14	8
Hill-Gregory	Positive	12	9
Hill-Gregory	Commercial	1	10
Young Inc	Scene	100	1
Young Inc	Wife	94	2
Young Inc	Drop	81	3
Young Inc	Order	69	4
Young Inc	Window	64	5
Young Inc	Official	33	6
Young Inc	Letter	28	7
Young Inc	Positive	28	7
Young Inc	Let	23	9
Young Inc	Still	15	10
Hoffman PLC	Identify	98	1
Hoffman PLC	Begin	89	2
Hoffman PLC	Mrs	87	3
Hoffman PLC	Cultural	87	3
Hoffman PLC	Onto	70	5
Hoffman PLC	Possible	62	6
Hoffman PLC	Never	47	7

5. Most Active Users by Post Count

```
SELECT
    u.user_id,
    u.fname,
    u.lname,
    COUNT(p.post_id) as post_count
FROM [user] u
LEFT JOIN post p ON u.user_id = p.created_by_user_id
GROUP BY u.user_id, u.fname, u.lname
ORDER BY post_count DESC;
```

Relational Algebra:

$\pi_{\text{user_id}, \text{fname}, \text{lname}, \text{post_count}} (\gamma_{\text{user_id}, \text{fname}, \text{lname}}; \text{COUNT}(\text{post_id}) \rightarrow \text{post_count}$
 $(\text{user} \bowtie [\text{user_id} = \text{created_by_user_id}] \text{post}))$

output:

	user_id	fname	lname	post_count
1	1066	Jennifer	Gallagher	80
2	1067	Cheyenne	Howard	80
3	1068	Tara	Carter	80
4	1069	Melissa	Yang	80
5	1070	Maria	Ward	80
6	1071	Brandon	Santiago	80
7	1072	Debra	Walker	80
8	1073	Valerie	Spence	80
9	1074	Lisa	Pitts	80
10	1075	Teresa	Mann	80
11	1076	Mason	Romero	80
12	1077	Donna	Scott	80
13	1078	Tammy	Rivera	80
14	1079	Max	Sims	80
15	1080	James	Johnson	80
16	1081	Vanessa	Moore	80
17	1082	Cheyenne	Welch	80
18	1083	Martin	Baxter	80
19	1084	Rebecca	Hinton	80

6. Most Popular Product Categories

```
SELECT
    ba.category,
    COUNT(p.product_id) as total_products,
    AVG(p.price) as avg_price
FROM business_account ba
JOIN product p ON ba.business_id = p.business_id
GROUP BY ba.category
ORDER BY total_products DESC;
```

Relational Algebra:

$\pi_{category, total_products, avg_price} (\gamma_{category; COUNT(product_id) \rightarrow total_products, AVG(price) \rightarrow avg_price} (business_account \bowtie [business_account.business_id = product.business_id] product))$

output:

	category	total_products	avg_price
1	cultivate customized channels	150	492.033933
2	seize scalable portals	110	423.336363
3	transition virtual convergence	100	493.522100
4	transition plug-and-play deliverables	80	536.571875
5	visualize bricks-and-clicks supply-chains	80	450.662000
6	visualize web-enabled platforms	80	523.136250
7	whiteboard customized applications	80	491.097750
8	visualize one-to-one infrastructures	80	547.895500
9	whiteboard innovative architectures	80	479.019500
10	whiteboard next-generation partnerships	80	514.776375
11	synergize real-time e-services	80	490.381750
12	synergize revolutionary solutions	80	503.988250
13	synthesize B2B partnerships	80	489.524875
14	synthesize B2C interfaces	80	535.396875
15	synthesize cross-media e-business	80	435.864875
16	synthesize strategic solutions	80	531.938625
17	synthesize turn-key e-tailers	80	504.560000
18	target dynamic networks	80	497.111375
19	target leading-edge interfaces	80	475.382750

7. Business Account Performance Metrics

```
SELECT
    ba.business_name,
    COUNT(DISTINCT f.following_user_id) as followers,
    COUNT(DISTINCT p.post_id) as total_posts
FROM business_account ba
JOIN [user] u ON ba.user_id = u.user_id
LEFT JOIN follower f ON u.user_id = f.followed_user_id
LEFT JOIN post p ON u.user_id = p.created_by_user_id
LEFT JOIN reaction r ON p.post_id = r.post_id
GROUP BY ba.business_name
ORDER BY followers DESC;
```

Relational Algebra:

$\pi_{\text{business_name}, \text{followers}, \text{total_posts}} (\gamma_{\text{business_name}; \text{COUNT}(\text{DISTINCT} \text{following_user_id}) \rightarrow \text{followers}, \text{COUNT}(\text{DISTINCT} \text{post_id}) \rightarrow \text{total_posts}} (\text{business_account} \bowtie_{[\text{business_account.user_id} = \text{user.user_id}]} \text{user} \bowtie_{[\text{user.user_id} = \text{follower.followed_user_id}]} \text{follower} \bowtie_{[\text{user.user_id} = \text{post.created_by_user_id}]} \text{post} \bowtie_{[\text{post.post_id} = \text{reaction.post_id}]} \text{reaction}))$

Output:

	business_name	followers	total_posts
1	Glover-Lewis	4	80
2	Hawkins LLC	4	70
3	Kelley, Holmes and Schwartz	4	70
4	Adkins, Simmons and Frazier	3	80
5	Clark-Carter	3	80
6	Dixon, Murphy and Mccarthy	3	60
7	Pacheco Group	3	80
8	Watkins, Phillips and Sullivan	3	80
9	Wagner Inc	2	80
10	Wagner, Kelly and Rodriguez	2	80
11	Shaw Inc	2	80
12	Shelton, Mcdonald and Guzman	2	80
13	Smith-Escobar	2	80
14	Sullivan PLC	2	60
15	Summers, Sutton and Trujillo	2	80
16	Tucker, Mendez and Bryan	2	60
17	Oneill-Russell	2	80
18	Nielsen Ltd	2	80
19	Nixon LLC	2	70

8. Content Type Performance Analysis

```
SELECT
    pt.post_type_name,
    COUNT(DISTINCT p.post_id) as total_posts,
    COUNT(DISTINCT r.user_id) as total_reactions,
    COUNT(DISTINCT c.id) as total_comments
FROM post_type pt
JOIN post p ON pt.id = p.post_type
LEFT JOIN reaction r ON p.post_id = r.post_id
LEFT JOIN comment c ON p.post_id = c.post_id
GROUP BY pt.post_type_name;
```

Relational Algebra:

π post_type_name, total_posts, total_reactions, total_comments(γ
post_type_name;COUNT(DISTINCT post.post_id) \rightarrow total_posts,COUNT(DISTINCT
reaction.user_id) \rightarrow total_reactions,COUNT(DISTINCT comment_id) \rightarrow total_comments
(post_type \bowtie post \bowtie reaction \bowtie comment))

	post_type_name	total_posts	total_reactions	total_comments
1	post	27047	48	74
2	reel	27032	48	54
3	story	26921	49	72

9. Geographic Distribution of Posts

```
SELECT
    p.location,
    COUNT(DISTINCT p.post_id) as total_posts,
    COUNT(DISTINCT r.user_id) as total_engagement
FROM post p
LEFT JOIN reaction r ON p.post_id = r.post_id
WHERE p.location IS NOT NULL
GROUP BY p.location
HAVING COUNT(DISTINCT p.post_id) > 5
ORDER BY total_posts DESC;
```

Relational Algebra:

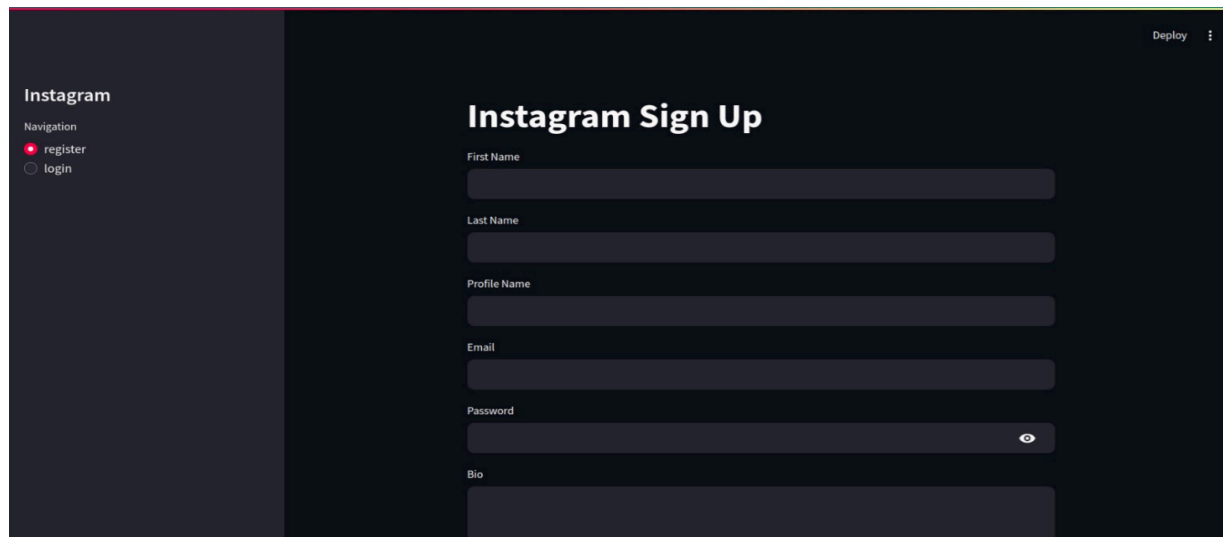
$\sigma_{\text{total_posts} > 5}(\pi_{\text{location, total_posts, total_engagement}}(\gamma_{\text{location}; \text{COUNT}(\text{DISTINCT post.post_id}) \rightarrow \text{total_posts}, \text{COUNT}(\text{DISTINCT reaction.user_id}) \rightarrow \text{total_engagement}}(\sigma_{\text{location IS NOT NULL}}(\text{post} \bowtie \text{reaction}))))$

Output:

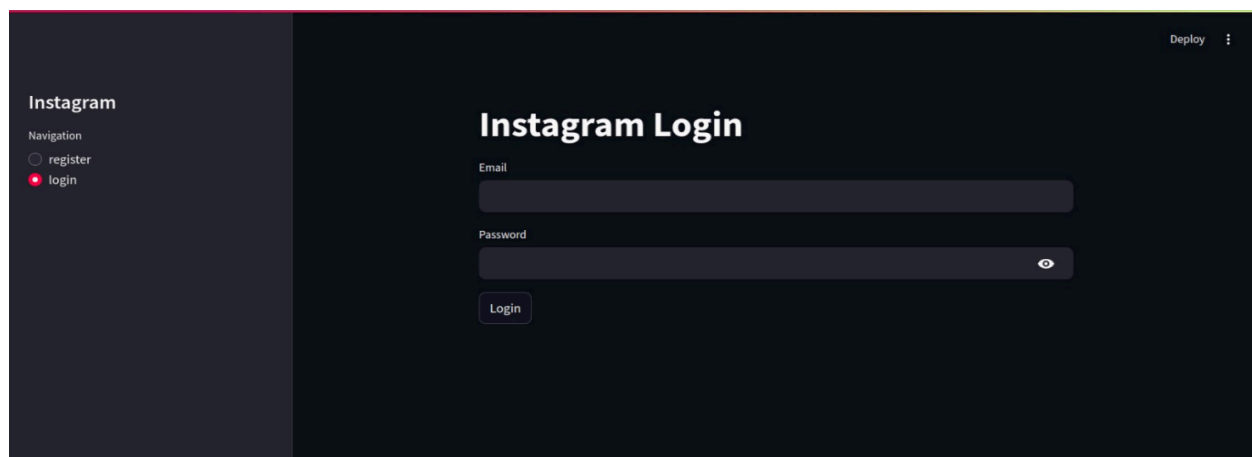
	location	total_posts	total_engagement
1	East Michael	72	0
2	West Michael	66	1
3	South Michael	65	0
4	Lake Michael	62	0
5	Michaemouth	60	0
6	New Michael	56	0
7	Port Michael	54	0
8	New Christopher	52	0
9	South John	51	0
10	Lake Jennifer	51	0
11	East David	51	0
12	West James	51	0
13	North Michael	50	0
14	New James	49	0
15	Port John	49	0
16	Port James	48	0
17	South James	48	0
18	New John	48	0
19	North Robert	48	0

#Bonus

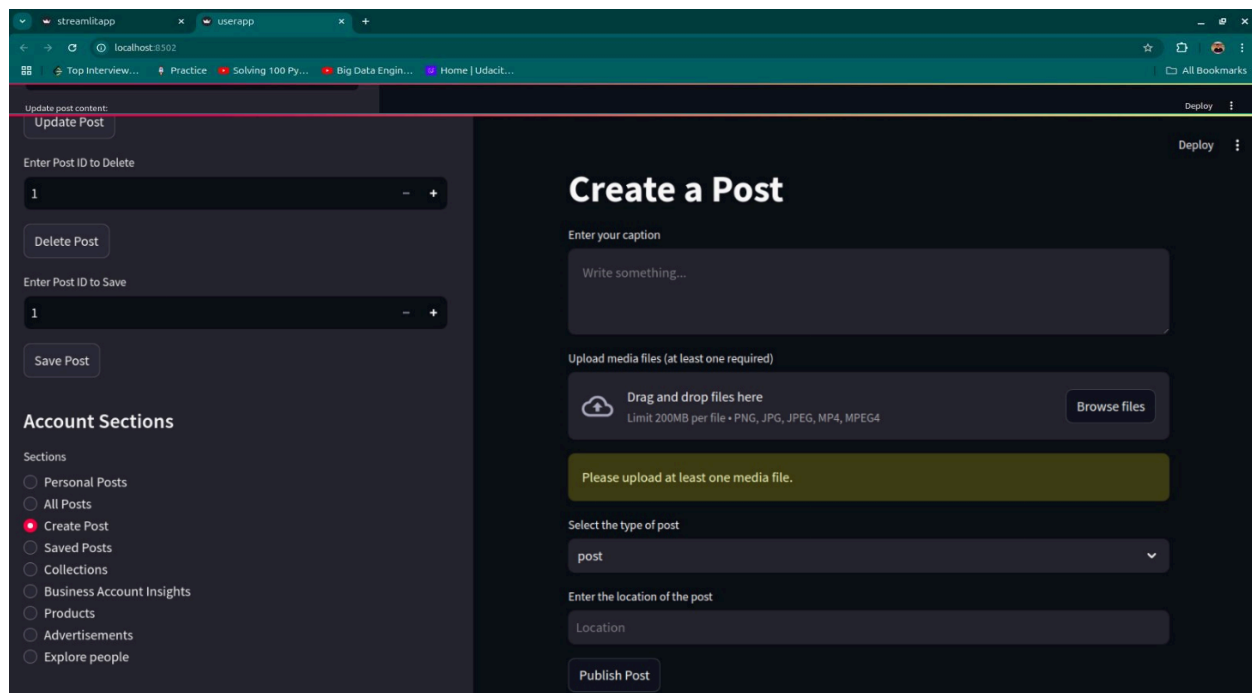
- GUI



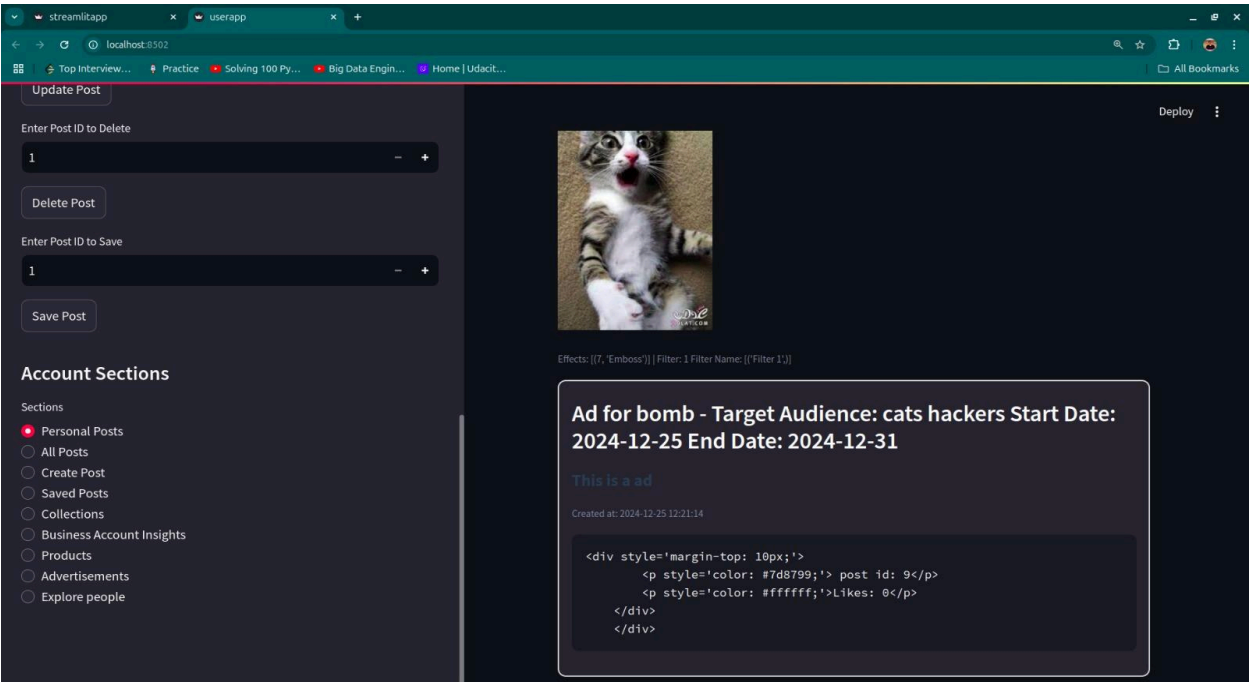
The image shows a dark-themed web application interface for an Instagram sign-up page. On the left, a sidebar contains the title 'Instagram' and a 'Navigation' section with two radio buttons: 'register' (selected) and 'login'. The main content area is titled 'Instagram Sign Up' and features a 'Deploy' button in the top right corner. The form includes input fields for 'First Name', 'Last Name', 'Profile Name', 'Email', 'Password' (with a toggle icon), and 'Bio'.



The image shows the same dark-themed web application interface, but for the login page. The sidebar now has the 'login' radio button selected. The main content area is titled 'Instagram Login' and includes a 'Deploy' button. The form consists of 'Email' and 'Password' input fields (the password field has a toggle icon) and a 'Login' button.



The image shows a browser window with two tabs: 'streamitapp' and 'userapp'. The 'userapp' tab is active, displaying a 'Create a Post' form. The browser's address bar shows 'localhost:8502'. The sidebar on the left has 'Create Post' selected under the 'Account Sections' menu. The main form area has a 'Deploy' button and includes fields for 'Update post content' (with an 'Update Post' button), 'Enter Post ID to Delete' (with a 'Delete Post' button), and 'Enter Post ID to Save' (with a 'Save Post' button'). The 'Create a Post' section includes a caption field, a media upload area with a 'Browse files' button, a message 'Please upload at least one media file.', a dropdown for 'Select the type of post' (set to 'post'), a 'Location' field, and a 'Publish Post' button.



#	Full Name	ID	Dept	SEC#	M#1	M#2	M#3	M#4	Bonus	Total
1	Omar Mohamed Ahmed Mashaly	320220079	AID	1						
2	Seif Eldeen Sameh	320220062	AID	1						
3	Omar Essameldin Mousa	320220022	AID	1						
4	Mostafa Mohamed Atef	320220034	AID	1						
5	Ahmed Mohamed Moner	320210176	AID	1						
6										

Total