Social Feed Backend - Submission Package

1. SQL Schema (database.sql)

```
DROP DATABASE IF EXISTS social_feed;
CREATE DATABASE social_feed CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
USE social_feed;

CREATE TABLE users ( id INT
AUTO_INCREMENT PRIMARY KEY, username
VARCHAR(100) NOT NULL UNIQUE, email
VARCHAR(150) NOT NULL UNIQUE, password
VARCHAR(255) NOT NULL,
role ENUM('user','admin','owner') DEFAULT 'user',
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE posts ( id INT AUTO_INCREMENT
PRIMARY KEY, user_id INT NOT NULL, content TEXT,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);

CREATE TABLE likes ( id INT
AUTO_INCREMENT PRIMARY KEY,
user_id INT NOT NULL, post_id INT NOT
NULL,

created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
UNIQUE KEY uniq_like (user_id, post_id),
FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
FOREIGN KEY (post_id) REFERENCES posts(id) ON DELETE CASCADE
);

CREATE TABLE follows ( id INT AUTO_INCREMENT
PRIMARY KEY, follower_id INT NOT NULL, following_id INT
NOT NULL, created_at TIMESTAMP DEFAULT
CURRENT_TIMESTAMP,
UNIQUE KEY uniq_follow (follower_id, following_id),
FOREIGN KEY (follower_id) REFERENCES users(id) ON DELETE CASCADE,
FOREIGN KEY (following_id) REFERENCES users(id) ON DELETE CASCADE
);

CREATE TABLE blocks ( id INT AUTO_INCREMENT
PRIMARY KEY, blocker_id INT NOT NULL, blocked_id INT
NOT NULL, created_at TIMESTAMP DEFAULT
CURRENT_TIMESTAMP,
UNIQUE KEY uniq_block (blocker_id, blocked_id),
FOREIGN KEY (blocker_id) REFERENCES users(id) ON DELETE CASCADE,
FOREIGN KEY (blocked_id) REFERENCES users(id) ON DELETE CASCADE
);

CREATE TABLE activity ( id INT
AUTO_INCREMENT PRIMARY KEY, type
ENUM('post','follow','like','user_deleted','pos
t_deleted','block','user_role_change') NOT
```

```
NULL, actor_id INT, target_user_id INT,
post_id INT, meta JSON NULL, created_at
TIMESTAMP DEFAULT
CURRENT_TIMESTAMP,
FOREIGN KEY (actor_id) REFERENCES users(id) ON DELETE SET NULL,
FOREIGN KEY (target_user_id) REFERENCES users(id) ON DELETE SET NULL,
FOREIGN KEY (post_id) REFERENCES posts(id) ON DELETE SET NULL
);
```

2. Postman Export Instructions

- Create a new collection in Postman.
- Add requests: signup, login, create post, like, follow, block, make-admin, revoke-admin, delete user.
- Click the three dots next to the collection → Export → Collection v2.1 → Save as postman_collection.json.

3. Deployment Note

Local deployments (http://localhost:4000) are accepted if you do not have online hosting.