

# Instagram Feed Clone

Name: Namburu Mohan Krishna

Roll.no: 126156091

**Github Repo:** [https://github.com/mohannamburu18/instagram\\_feednew](https://github.com/mohannamburu18/instagram_feednew)

**Deployment link:** <https://instagram-feednew-jc9l6pudq-mohans-projects-7ac7513d.vercel.app/>

## 1. Track Chosen + Why (2-3 lines)

### Track C Feed (Instagram type consumption)

I selected Track C, which is used to show an interactive feed with pagination and modal post reading, as well as social features such as like/save. This track captures UX of the frontend, API integration and state management in a production-oriented manner in the best way possible.

## 2. Aspects Implemented (Checklist)

### Frontend (UI)

- Instagram-style feed grid
- Post commenter (desktop and mobile)
- Next / Previous post navigation
- Like toggle with live count update
- Bookmark toggle using localStorage
- Swipe (up/down) navigation on mobile
- Mobile floating action buttons (like / save)
- Fully responsive design (desktop & mobile)
- Empty state and loading state handling

- Error handling UI

### **Backend (API)**

- REST API using Express
- Get posts (pagination)
- Like a post
- Create / Delete post
- Validation + error responses

### **Persistence**

- SQLite database for posts
- localStorage for likes & saved posts

## **3. Tech Stack**

### **Frontend**

- React (Vite)
- CSS (custom, responsive)
- Axios

### **Backend**

- Node.js
- Express.js
- SQLite

### **Deployment**

- Frontend: Vercel
- Backend: Render

#### **4. How to run:**

1. Clone the repository from GitHub using  
`git clone https://github.com/mohannamburu18/instagram_feednew.git`
2. Move into the project directory  
`cd instagram_feednew`
3. Navigate to the backend folder  
`cd server`
4. Install backend dependencies  
`npm install`
5. Create a .env file inside the server folder and add `PORT=5000`
6. Start the backend server  
`npm run dev` or `npm start`
7. Open a new terminal window
8. Navigate to the frontend folder  
`cd client`
9. Install frontend dependencies  
`npm install`
10. Create a .env file inside the client folder and add  
`VITE_API_URL=http://localhost:5000/api`
11. Restart the frontend server  
`npm run dev`
12. Open the application in the browser at <http://localhost:5173>

#### **5.API Endpoints**

<b>Method</b>	<b>Endpoint</b>	<b>Description</b>
GET	/api/posts?page=1&limit=12	Get paginated posts
POST	/api/posts/:id/like	Like a post
POST	/api/posts	Create new post
DELETE	/api/posts/:id	Delete post
GET	/api/health	Health check

## **6. Data Model (Tables / Fields)**

### **posts table**

Field	Type
id	INTEGER (PK)
author	TEXT
caption	TEXT
image	TEXT
likes	INTEGER
creator_id	TEXT
created_at	TIMESTAMP

## **AI Collaboration Log**

### **AI Tools Used**

- ChatGPT
- Claude
- Gemini

### **How AI Helped**

- Gemini helped in gathering visual references and overall design inspiration.
- ChatGPT assisted with deployment and hosting guidance.
- Suggested state management for like / save toggle functionality.
- Improved component structure, organization, and UI visibility.

## Example Prompt

- Trendy React Instagram-like feed with grid of posts and fullscreen post viewer with next/previous navigation.
- My posts do not load on Vercel; the API request waits indefinitely. Can you help debug frontend vs backend issues?

## Problem

- Images overlapped with the right sidebar.
- Background feed remained scrollable.
- Next / Previous buttons were not consistent.
- Viewer was embedded instead of fullscreen.

## Your Correction

- Added a fullscreen overlay layer.
- Locked background scrolling when the viewer is open.
- Separated the layout into:
  - viewer-media (left – image)
  - viewer-sidebar (right – author, caption, actions)

```
useEffect(() => {
  document.body.style.overflow = viewerOpen ? 'hidden' : 'auto';
}, [viewerOpen]);
```

## 8.Trade-offs

- LocalStorage is used instead of a backend for saved posts.
- Authentication is mocked using LocalStorage.
- Comments are UI-only with no backend persistence.

### Next Improvements

- Add a dedicated Saved Posts page.

.

- Connect comments and likes to a backend service.

## **9. Sample Data**

- Data is stored using SQLite seed data.
- Photos are loaded using fixed URLs.

## **10. Demo Video**

[https://drive.google.com/file/d/1Df1oazf72RH6\\_KPR6UJ5EZgXff9e69eM/view?usp=sharing](https://drive.google.com/file/d/1Df1oazf72RH6_KPR6UJ5EZgXff9e69eM/view?usp=sharing)