Postdost_Development_Docs

Initial Setup:

1. Clone the Github repository to your local storage
2. Checkout to your assigned branch
3. Use the initial folder structure setup to create or edit files
   - ○ Must Follow the **Coding patterns** described below (otherwise leads to complexity and inconsistency)
4. Use `npm run dev` to start the development server as it was a **vite** created react directory.
5. commit and push to your branch on the remote repo
6. Finally when asked, publish a pull request to merge your branch into main branch

NOTE: I have created dummy Ui components for you to start development, we can always fix the styling and html later.

For now just develop the project structure and components needed.

Tech Stack:
1. Node JS
2. Express JS (Back-end node framework)
3. React JS (Front-end Framework)
4. Tailwind CSS/UI , Chakra UI, Modular CSS (which ever you prefer)
5. MongoDB (Database)
6. GitHub/Git (for Version control and branching)
7. ClickUp (For project management, responsibilities and tasks)
8. Whatsapp Group for all communication

Coding practices you must follow:

1. Use UI Components in components/UI directory for all reusable elements e.g. Buttons, cards etc.
2. You can Customize the UI elements in components/UI but only change the CSS.

3. Follow **Single Responsibility Principle** i.e. Each functional or non-functional component should have one activity or role to perform
4. Create Basic Reusable Elements using styled-components library i.e. Buttons, Cards, Fields, Boxes etc.
5. Use `Context API` for all state management in React. Use contexts directory for this.
6. Create separate components for functions and static elements
7. Always use Fragments i.e. `<></>`
8. **Modular CSS** is preferred for all styling
9. **Use Variables for fonts, colors, reusable styles in CSS**
10. Create separate folder for each Page
11. Use a RESTful API design pattern for the server-side API. This includes using HTTP verbs (GET, POST, PUT, DELETE) to perform CRUD operations and following a consistent API endpoint naming convention
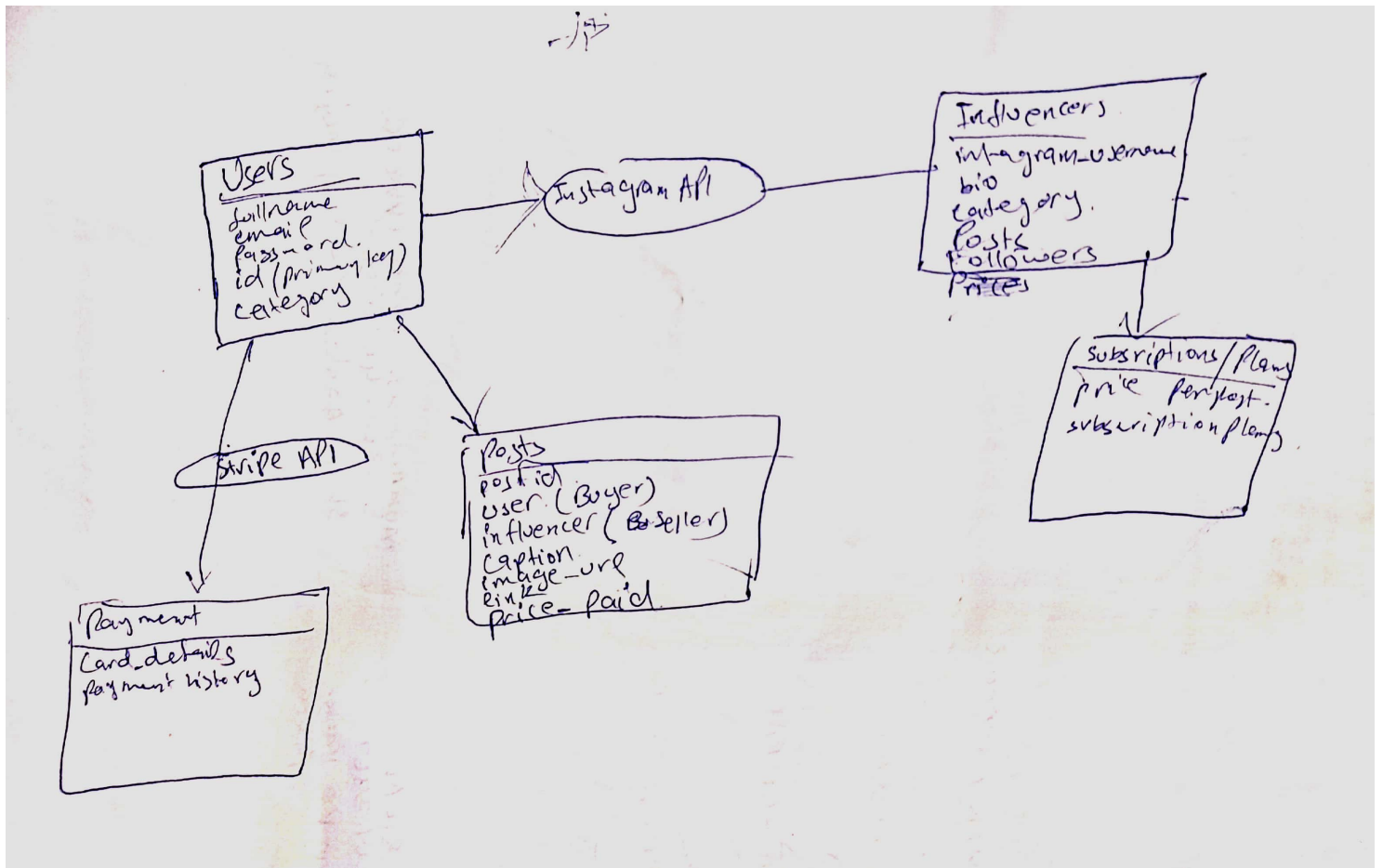
Responsibilities and Tasks:

All your tasks would be assigned and tracked on your branch list on ClickUp Space.
Following are the branches assigned so please only work on your own branch:

1. client-side-u: @Usama Kamran
2. server-side-e: @Ehsan Ellahi
3. server-side-m: @Mayar
4. main: @Ammar Qureshi

Data Schema:

Users
- fullname
- email
- password
- id (Primary key)
- category

Instagram API

Influencers
- instagram-username
- bio
- category
- posts
- followers
- prices

Subscriptions/Plans
- price per post
- subscription plans

Stripe API

Posts
- post id
- user (Buyer)
- influencer (Re-Seller)
- caption
- image_url
- link
- price-paid

Payment
- card-details
- payment history

Links:

Figma:

Github:
https://github.com/ammarrqureshi/postdost-web-app

Guide Video: