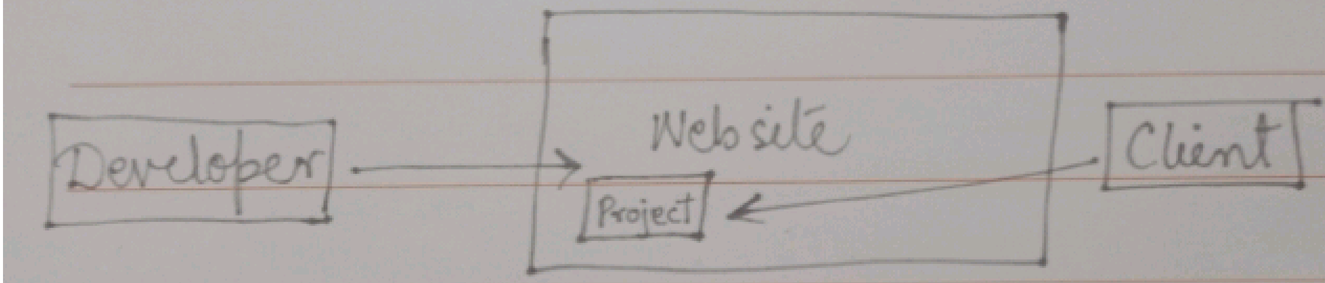
"Next Js
Marketplace
2025"

"IDEA"

Step # 1 : Market Type

"General E-commerce"



"my platform is going to connect developers and their unseen projects with clients of their projects neeche it is basically the same work we do on upwork & github separatly"

* Key components of my website :

1- Showcase Platform for developers :

* developers upload and showcase their project (ex: web apps, mobile apps, APIs. etc)
* Buyers browse and purchase ready made projects on their interest.

*

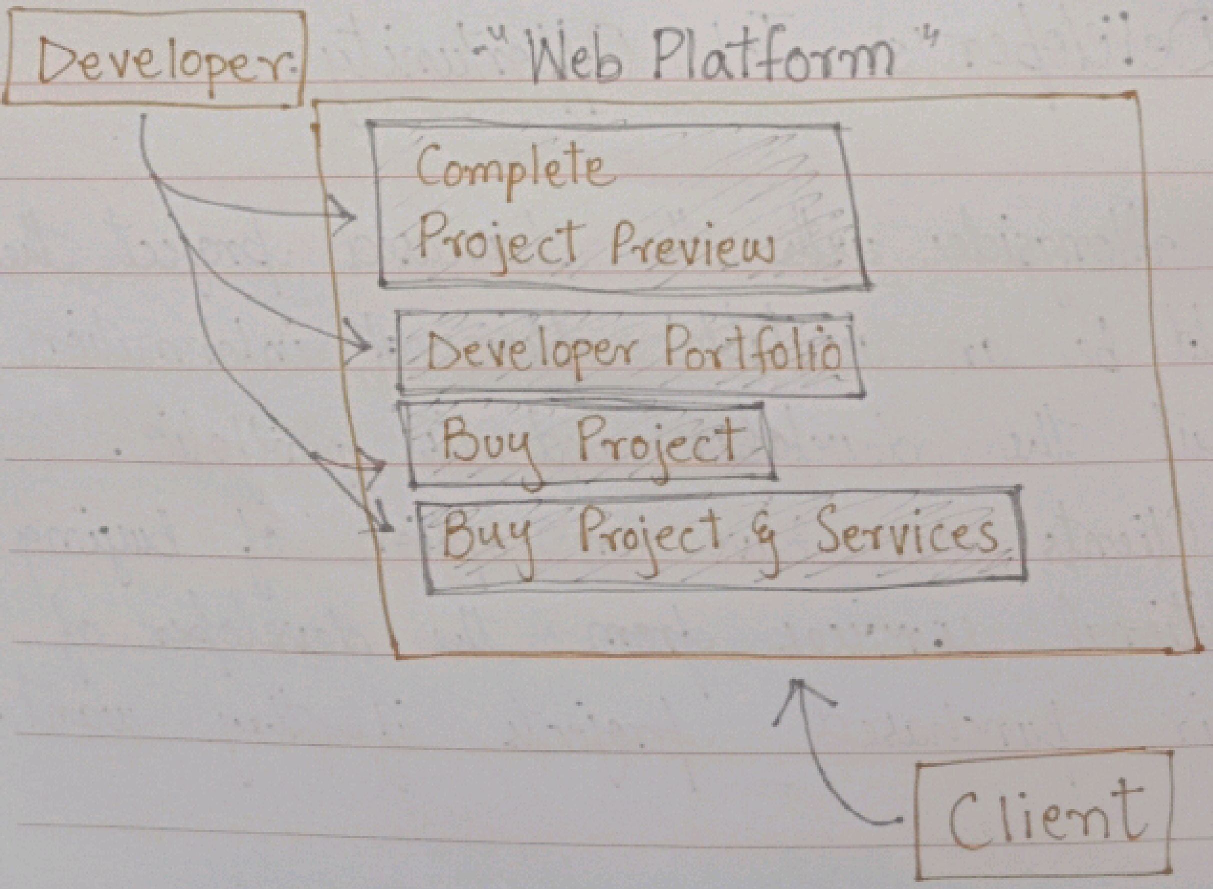
2.- Developer's Job Opportunity :

* Alongside with the showcased project their would be an available option of information about the developer and its portfolio.
* Clients can have an option of buying additional services from the developer of their purchased projects if they want.

3- Dual Marketplace:

* Product Marketplace: Focused on selling complete projects.

* Services Marketplace: Optionally connects buyers with developers for post-purchase needs.



STEP # 2: GOAL'S :

Q) What problem is being solved ??

A project marketplace (like CodeCanyon) & a developor marketplace (like Upwork or Fiverr) at same place providing Seamless integration of optional post-purchase services and live project preview which existing Platform lack, decision making for clients would be easier.

Q) Target Audience ?

People providing services & people needing services, both at once.

STEP # 3: DATA SCHEMA :

~~Products~~ ★ USERS: represents both developer & client

- id (string)
- name
- email
- password.
- profile pic
- role (developer | client)
- bio
- skills (optional for clients)

★ Projects: uploaded by developers

- id
- developer Id (refrering to user id)
- tags (New arrival etc)

- title
- description
- category (mobile app etc)
- price
- Image Urls (array)
- created At
- availibility (purchased or not)

★ Services : optional offered by developers

- id
- developer Id (refering to user id)
- name
- description
- price (per hour or contract base)
- available (boolean)

* Transactions : purchase made by client.
either (project or services.)

- id
- buyer Id (reference to user id)
- Items Id (referring to project id or service id)
- item type (project & services | project | services)
- amount
- status (pending | completed | failed)

"RELATIONSHIP"

- User → has many → Projects & Services
- Client (User) → has many → Transactions
- Developer (User) → offers many → Projects and Services