# Capstone Project Technical Barriers Assessment

## Name of Project:

## Lead Developer: **Himani Bansal**

## Potential Project Barriers

### Content

1. *Finding Quality Images*  
   It might be hard to find high-quality images of products that are free to use. I may also need to edit or resize them to fit the website's layout and performance needs.
2. *Writing Product Descriptions*  
   Writing clear and unique product descriptions takes time. They also need to be short, interesting, and good for search engines (SEO).
3. *Accurate Pricing and Categorization*

It can be tricky to figure out fair prices for handmade or special products. Some items might fit into more than one category, like 'Plushies' being both toys and personalized items.

### Content Barriers – Mitigation Plan

I will use free image platforms like Canva to get good-quality, copyright-free images. Canva also helps edit and resize images easily. For product descriptions, I will follow simple templates to keep things clear and consistent. I might use tools like Grammarly or ChatGPT to help write drafts and then edit them myself to make sure they sound original. For prices and categories, I will check similar websites like Etsy or Amazon and use a clear list to keep items organized and avoid confusion.

### Technology Stack

4. *New to Full-Stack Web Development*  
Learning to build the frontend, backend, and database all together can feel overwhelming since I haven’t done it before.

5. *Node.js + Express*  
It may be hard to understand how asynchronous code works or how to set up middleware and routes properly.

6. *React (or plain HTML/CSS/JS)*  
React might be difficult because I need to learn how to use components, props, state, and hooks.

### Technology Barriers – Mitigation Plan

To manage this, I’ll build the frontend, backend, and database one step at a time. I’ll first test each part separately and then connect them. For Node.js and Express, I’ll review what I’ve done in class and practice building small pieces like login forms or routes. I will use Postman to test backend APIs. If React is too hard at first, I’ll build the layout using HTML, CSS, and JavaScript, then move to React later. To save time, I’ll use Bootstrap for styling and rely on online tutorials and forums when I get stuck.

#### 3rd Party Libraries and/or Frameworks

7. *Stripe or PayPal – nice to have*  
I’ve never worked with online payments before, so I need to learn how to add it to my site.

### API Barriers – Mitigation Plan

I will make developer accounts on Stripe and PayPal. This will give me access to sandbox mode, where I can test payments without using real money. I’ll get the test API keys from their dashboards and follow the official guides and sample code to set up a basic checkout page. Users will be able to make pretend purchases using test card numbers (for Stripe) or sandbox accounts (for PayPal). I’ll keep the keys safe in a .env file, and test both successful and failed payments. Since I'm only using sandbox mode, I don’t need to worry about real transactions. This makes it easier and safer to practice and learn.

#### APIs

**API NAME: Stripe**

**API URL:** <https://stripe.com/docs/api>

**Data:**

* Customer info (name, email)
* Payment method (card token)
* Transaction details (amount, currency, status)

**Requirements:**

* Requires registration at [dashboard.stripe.com](https://dashboard.stripe.com)
* You’ll need both publishable key and secret key
* Uses test card numbers in sandbox mode

**Cost/Limitations:**

* **Free** in sandbox mode
* No real transactions occur
* No rate limits for testing, but production has rate limits (e.g., 100 reads/sec per account)

**API NAME: PayPal Sandbox**

**API URL:** <https://developer.paypal.com/docs/api/overview/>

**Data:**

* Buyer email & approval
* Payment intent
* Transaction response

**Requirements:**

* Requires a PayPal Developer account
* Create sandbox buyer and merchant accounts
* Set up Client ID and Secret

**Cost/Limitations:**

* Free in sandbox
* Rate limits apply for production (e.g., ~1500 calls/hour)
* Limited to test users in sandbox