# Capstone Project Technical Barriers Assessment

## Name of Project: PetCake Delight

## Lead Developer: Peiyu Han

## Potential Project Barriers

### Content

* **Image Assets**: High-quality images of pet cakes, treats, and bakery interiors need to be sourced or created. Using images found online may lead to copyright issues.
* **Text Content**: Product descriptions, the About page, and FAQs must be original, engaging, and SEO-friendly, which takes time to draft and review.

**Mitigation Strategies**:

* Use royalty-free image libraries such as [Unsplash](https://unsplash.com) or [Pexels](https://www.pexels.com).
* Consider generating placeholder images using AI tools like DALL·E or Adobe Firefly until final assets are produced.
* Draft content early and use tools like Grammarly and DeepL for faster editing.
* Set up a simple content workflow: draft → review → finalization.

### Technology

#### Stack

* **WordPress and WooCommerce**: Though beginner-friendly, deep customization (e.g., modifying product templates or PHP-based functions) can be technically demanding.
* **WooCommerce Product Logic**: Implementing product variation (e.g., flavor, size, custom messages) and personalization logic may require understanding the WooCommerce data model.

**Mitigation Strategies**:

* Allocate early development time for reading documentation and watching tutorials on WooCommerce customizations.
* Use a local development environment (e.g., LocalWP) for safe experimentation.
* Leverage WP debugging tools (e.g., Query Monitor) and the WordPress developer forums for troubleshooting.

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

* **Plugins & Extensions**: Extending WooCommerce through plugins (e.g., product customization, email automation) may introduce conflicts, bugs, or security vulnerabilities.
* **CSS Frameworks**: If using Tailwind CSS or Bootstrap for styling, integration with WordPress themes could be non-trivial, especially if overriding theme defaults.

**Mitigation Strategies**:

* Select only essential, well-maintained plugins with strong reviews and documentation.
* Limit the number of plugins to reduce complexity and potential conflicts.
* Consider child themes or theme builders (e.g., Elementor) to ease layout design.
* Conduct plugin compatibility testing before final integration.

#### APIs

No external APIs currently planned.

API NAME

API URL

Data

Requirements (API key, registration…)

Cost/Limitations (subscription or per-use fee? # of calls per ?)