◼ Insta Downloader App – Project Specification Document

📌 Project Overview

The **Insta Downloader App** is designed to allow users to download Instagram content seamlessly. The app features a clean and intuitive interface, enabling users to input Instagram URLs and download photos, videos, or reels directly to their devices.

🎨 UI Design Summary

- **Design Tool:** Figma
- **Total Screens:** Approximately 10
- **Design Style:** Minimalistic and user-friendly
- **Color Scheme:** Light mode with accent colors
- **Typography:** Modern sans-serif fonts(Behance)

📱 Key Features

- **URL Input:** Users can paste Instagram post URLs to initiate downloads.
- **Content Preview:** Displays a preview of the content before downloading.
- **Download Options:** Allows downloading of images, videos, and reels.
- **Download History:** Maintains a history of downloaded content for user reference.
- **Settings:** Options to manage download preferences and app settings.(https://mycaptain.in/)

🛠 Technical Stack

- **Frontend:** Flutter (Dart)
- **Backend:** Laravel (PHP)
- **Database:** MySQL
- **API Integration:** Utilizes Instagram's public APIs or scraping methods, adhering to their terms of service.
- **Authentication:** Optional user authentication for personalized features.
- **Storage:** Local device storage for downloaded content.(GUVI, Dribbble)

🎁 Deliverables

- Complete Flutter mobile application (Android & iOS)
- Laravel backend with necessary APIs
- Database schema and setup scripts
- Comprehensive documentation
- Deployment guide

⊞ Estimated Timeline

Phase	Duration
Requirement Analysis	2 Days
UI Implementation	3 Days
Backend Development	4 Days
Integration & Testing	3 Days
Deployment	2 Days
Total Duration	**14 Days**

💰 Pricing Options

Package	Description	Price (USD)
Basic	UI implementation only (no backend)	$100
Standard	App with basic backend features	$300
Premium	Full-featured app with backend and deployment	$600

📎 Additional Notes

- Ensure compliance with Instagram's terms of service when implementing download functionalities.
- Consider implementing rate limiting and error handling for API interactions.
- Optional features like user authentication can enhance user experience but are not mandatory.