# Instagram Comment Generator – Design Document

## Overview

The Instagram Comment Generator is a web application that automates the creation of Instagram comments using AI. The platform utilizes a Next.js 14 front end, a Python FastAPI backend, and integrates with LLM services to generate responses based on Instagram post content.

## 1. Tech Stack

- Frontend: Next.js 14   
- Backend: Python with FastAPI  
- Web Scraping: Instaloader (for scraping Instagram posts)  
- LLM Integration: OpenRouter.ai (using Gemini 2.0 Pro model)  
- Deployment: Frontend on Vercel, Backend on Koyeb

## 2. Architecture

### Frontend (Next.js 14):

- React components for user interaction  
- TailwindCSS for styling  
- Axios for API requests

### Backend (FastAPI):

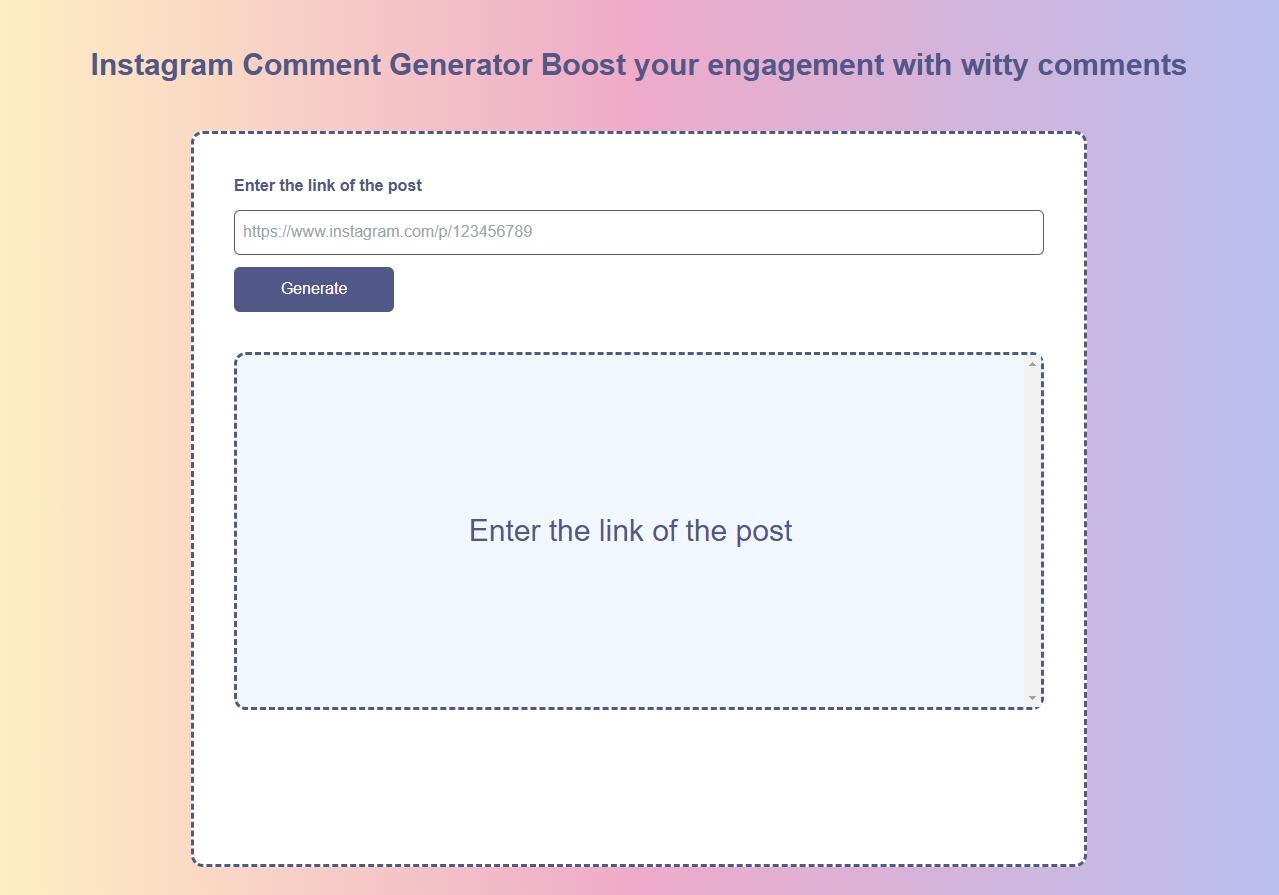
- API endpoints for receiving Instagram post data and returning generated comments  
- Web scraping with Instaloader  
- Integration with OpenRouter.ai for LLM responses

### LLM (OpenRouter.ai with Gemini 2.0 Pro):

- Prompt engineering for context-specific responses

## 3. Workflow

1. User Input: The user enters an Instagram post URL.  
2. Scraper: Backend scrapes the post content using Instaloader.  
3. LLM Query: The content is sent to OpenRouter.ai with a prompt to generate a suitable comment.  
4. Response: The generated comment is displayed to the user.



## 4. API Endpoints

- POST /generate-comment  
 - Input: Instagram Post URL  
 - Output: AI-generated comment

## 5. Security Considerations

- API Keys Management: Store keys in environment variables  
- Data Encryption: Use HTTPS for secure data transfer

## 6. Future Enhancements

- User Authentication: Enable sign-ins for saved comment history  
- Multi-model Support: Add support for different LLMs via OpenRouter.ai  
- Customization Options: Allow users to choose comment styles (funny, informative, promotional)

## 7. Dependencies

- Frontend: next, axios, tailwindcss  
- Backend: fastapi, uvicorn, instaloader, openai