FunnelCV Project Summary & Implementation Guide

Project Overview

FunnelCV is a simple web application that allows you to create tailored CV/resume summaries for specific job applications. It uses OpenAI's GPT to rewrite your CV summary for targeted roles, and generates a shareable link for each application.

Core Features

- Create personalized CV pages for specific job applications
- AI-powered summary tailoring using OpenAI API
- Video introduction embedding (YouTube/Loom)
- Mobile-optimized responsive design
- Persistent storage using Replit Database

Technical Requirements

- **Platform**: Replit (Flask template)
- **API**: OpenAI GPT-3.5-Turbo
- **Frontend**: HTML + Tailwind CSS
- **Database**: Replit DB for persistence
- **Monthly Budget**: Under $5 (estimated actual cost: <$1/month)

Implementation Steps

1. Project Setup

1. Create a new Replit project using the "Flask (Replit)" template
2. Name the project "funnelcv" or similar
3. Add your OpenAI API key to Replit Secrets with key name "OPENAI_API_KEY"

2. File Structure

```
/
├── main.py           # Main Flask application
├── templates/
│   ├── index.html    # Form for creating CVs
│   └── funnel.html   # Generated CV page template
├── .env              # Environment variables (via Replit Secrets)
└── poetry.lock       # Dependencies (auto-generated)
```

3. Code Implementation

main.py

python

```python
from flask import Flask, render_template, request, redirect
import openai
import os
from replit import db

app = Flask(__name__)

# Load OpenAI key from environment
openai.api_key = os.getenv("OPENAI_API_KEY")

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/generate', methods=['POST'])
def generate():
    name = request.form['name']
    job = request.form['job']
    company = request.form['company']
    summary = request.form['summary']
    video = request.form['video']

    # Prompt AI to rewrite summary (using cheaper GPT-3.5-Turbo)
    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",  # More cost-effective than GPT-4
        messages=[
            {"role": "system", "content": "Rewrite this CV summary for a " + job + " role at "
            {"role": "user", "content": summary}
        ]
    )

    rewritten = response.choices[0].message.content.strip()
    slug = f"{name.lower()}-{job.lower()}-{company.lower()}".replace(" ", "-")

    # Store in Replit DB for persistence
    db[slug] = {
        "name": name,
        "job": job,
        "company": company,
        "summary": rewritten,
        "video": video
    }

    return redirect(f'/cv/{slug}')

@app.route('/cv/<slug>')
```

```python
def funnel(slug):
    data = db.get(slug, None)
    if not data:
        return "Not found", 404
    return render_template('funnel.html', data=data)

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=81)
```

templates/index.html

```html
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Create Your Funnel CV</title>
  <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="style
</head>
<body class="p-6 bg-gray-50 text-gray-800">
  <form action="/generate" method="post" class="max-w-xl mx-auto space-y-4">
    <h1 class="text-2xl font-bold">Generate Your Funnel CV</h1>
    <input type="text" name="name" placeholder="Your Name" required class="w-full p-2 border rc
    <input type="text" name="job" placeholder="Job Title (e.g. UI Designer)" required class="w-
    <input type="text" name="company" placeholder="Company Name" required class="w-full p-2 bor
    <textarea name="summary" placeholder="Paste your CV summary here" required class="w-full p-
    <input type="url" name="video" placeholder="Link to your video (YouTube or Loom)" class="w-
    <button type="submit" class="bg-red-500 text-white px-4 py-2 rounded w-full md:w-auto">Gene
  </form>
</body>
</html>
```

templates/funnel.html

```html
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Funnel CV – {{ data.name }}</title>
  <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="style
</head>
<body class="p-4 md:p-6 bg-white text-gray-800 max-w-xl mx-auto">
  <h1 class="text-xl md:text-2xl font-bold mb-2">{{ data.name }} for {{ data.job }} at {{ data.

  {% if data.video %}
    <div class="relative pb-9/16 w-full h-0 mb-4" style="padding-bottom: 56.25%;">
      <iframe src="{{ data.video }}" class="absolute top-0 left-0 w-full h-full" frameborder="0
    </div>
  {% endif %}

  <h2 class="text-lg md:text-xl font-semibold mt-4 mb-2">Summary</h2>
  <p class="text-base">{{ data.summary }}</p>

  <div class="mt-6">
    <a href="mailto:kardos.sandor@gmail.com" class="block w-full md:w-auto text-center bg-greer
  </div>
</body>
</html>
```

Cost Optimization

1. **Using GPT-3.5-Turbo instead of GPT-4 Turbo**
   - Approximately 10x cheaper (~$0.001-$0.002 per CV generation vs ~$0.01-$0.02)
   - Still produces high-quality summaries for CV optimization
2. **Replit Free Tier**
   - Sufficient for the MVP and low to moderate usage
   - No additional hosting costs

Mobile Optimization

1. **Viewport Meta Tag** added to ensure proper scaling on mobile devices
2. **Responsive Design Elements**:
   - Full-width buttons on mobile, inline on desktop
   - Adjusted text sizes for mobile readability
   - Optimized video container for mobile viewing

- Proper padding for touch targets

Future Enhancements (Optional)

1. Add view tracking to see how many recruiters viewed your CV
2. Implement multiple CV sections (skills, experience, etc.)
3. Add authentication to allow multiple users
4. Create templates for different industries/roles
5. Add analytics for tracking engagement

Usage Instructions

1. Run the application on Replit
2. Fill out the form with your information:
   - Your name
   - Target job title
   - Company name
   - Your existing CV summary
   - Optional video link (YouTube/Loom)
3. Click "Generate Funnel Link"
4. Share the resulting link with recruiters

This project prioritizes simplicity, speed, and cost-effectiveness while still providing a functional and valuable tool for job applications.