**IELTS Study Plan Generator - Complete Documentation & Enhancement Plan**

**📋 Current App Analysis**

**Live App URL: https://ielts-app-1.onrender.com/**

**Current File Structure**

C:\projects\ielts-study-plan-generator\

├── .devcontainer/ # Development container config

├── \_\_pycache\_\_/ # Python cache files

├── assets/ # Static assets (images, styles, etc.)

├── app.py # Main Flask application (6,509 bytes)

├── enhancements.py # New enhancement features (29,472 bytes)

├── study\_plan\_logic.py # Core study plan generation logic (5,110 bytes)

├── minimal.py # Minimal version/testing (85 bytes)

├── test.ipynb # Jupyter notebook for testing (empty)

├── requirements.txt # Python dependencies (70 bytes)

├── README.md # Project documentation (4,344 bytes)

└── LICENSE # License file (1,085 bytes)

**🔍 Version Comparison Analysis**

**Online Version Features:**

* Simple form-based interface
* Basic study plan generation
* Limited styling and UX
* Focus on functional delivery

**Enhanced HTML Version Features:**

* **Professional branding** with SparkSkyTech logo integration
* **Modern UI/UX** with gradients, animations, and responsive design
* **Color-coded resource sections** for different IELTS skills
* **Comprehensive resource links** organized by categories
* **Multi-language support** (English/Arabic)
* **Advanced visual elements** (sparkle animations, hover effects)
* **Enhanced typography** with Google Fonts (Poppins)

**📁 Missing Files & Folders Analysis**

Based on your current structure, here are likely missing components:

**Frontend Assets**

/static/

├── css/

│ ├── style.css

│ └── responsive.css

├── js/

│ ├── main.js

│ ├── form-handler.js

│ └── language-switcher.js

├── images/

│ ├── logo/

│ │ ├── sparkskytech-logo.png

│ │ └── ielts-logo.png

│ └── icons/

│ ├── listening.svg

│ ├── reading.svg

│ ├── writing.svg

│ └── speaking.svg

└── fonts/

└── (if using local fonts)

**Backend Templates**

/templates/

├── base.html

├── index.html

├── study\_plan.html

├── error.html

└── partials/

├── header.html

├── footer.html

└── resource-links.html

**Configuration & Deployment**

├── config.py # App configuration

├── .env # Environment variables

├── .gitignore # Git ignore file

├── Procfile # Heroku/Render deployment

├── runtime.txt # Python version specification

└── gunicorn.conf.py # WSGI server config

**🚀 Enhancement Implementation Plan**

**Phase 1: Core Infrastructure Setup**

**1.1 File Structure Reorganization**

# Recommended structure

ielts-study-plan-generator/

├── app/

│ ├── \_\_init\_\_.py

│ ├── routes.py

│ ├── models.py

│ └── utils.py

├── static/

├── templates/

├── config/

├── tests/

└── deployment/

**1.2 Environment Configuration**

# config.py

import os

from dotenv import load\_dotenv

load\_dotenv()

class Config:

SECRET\_KEY = os.environ.get('SECRET\_KEY') or 'dev-secret-key'

DEBUG = os.environ.get('DEBUG', 'False').lower() == 'true'

LANGUAGES = {

'en': 'English',

'ar': 'العربية'

}

**Phase 2: Frontend Enhancement Integration**

**2.1 Template Structure Implementation**

Create modular templates that incorporate the enhanced HTML design:

<!-- templates/base.html -->

<!DOCTYPE html>

<html lang="{{ get\_locale() }}">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>{% block title %}IELTS Smart Study Plan Generator{% endblock %}</title>

<!-- Enhanced CSS from your HTML -->

<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" rel="stylesheet">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700;800&display=swap" rel="stylesheet">

<link rel="stylesheet" href="{{ url\_for('static', filename='css/enhanced-style.css') }}">

{% block head %}{% endblock %}

</head>

<body>

{% include 'partials/header.html' %}

<main class="main-content">

{% block content %}{% endblock %}

</main>

{% include 'partials/resources-section.html' %}

{% include 'partials/footer.html' %}

<script src="{{ url\_for('static', filename='js/main.js') }}"></script>

{% block scripts %}{% endblock %}

</body>

</html>

**2.2 Enhanced CSS Integration**

Extract and modularize the CSS from your HTML:

/\* static/css/enhanced-style.css \*/

/\* Copy the comprehensive CSS from your HTML file \*/

/\* Organize into sections: \*/

/\* - Base styles \*/

/\* - Header styles \*/

/\* - Form styles \*/

/\* - Resource section styles \*/

/\* - Responsive design \*/

**Phase 3: Backend Enhancement Integration**

**3.1 Enhanced Study Plan Generator**

# app/study\_plan\_generator.py

class EnhancedStudyPlanGenerator:

def \_\_init\_\_(self):

self.skill\_categories = {

'listening': {'color': 'skill-listening', 'icon': 'fas fa-headphones'},

'reading': {'color': 'skill-reading', 'icon': 'fas fa-book-open'},

'writing': {'color': 'skill-writing', 'icon': 'fas fa-pen-fancy'},

'speaking': {'color': 'skill-speaking', 'icon': 'fas fa-microphone'}

}

def generate\_enhanced\_plan(self, user\_data):

# Incorporate the enhanced logic from enhancements.py

# Add visual elements and categorization

pass

**3.2 Multi-language Support**

# app/translations.py

from flask\_babel import Babel, gettext, ngettext

def init\_babel(app):

babel = Babel(app)

@babel.localeselector

def get\_locale():

return request.args.get('lang', 'en')

**Phase 4: Advanced Features Implementation**

**4.1 Interactive Study Plan Dashboard**

* Progress tracking with visual indicators
* Skill-based color coding
* Interactive calendar view
* Progress analytics

**4.2 PDF Generation Enhancement**

# Enhanced PDF generation with styling

from weasyprint import HTML, CSS

from flask import render\_template

def generate\_styled\_pdf(study\_plan\_data):

html\_content = render\_template('pdf/study\_plan\_template.html',

data=study\_plan\_data)

css\_styles = CSS(filename='static/css/pdf-styles.css')

return HTML(string=html\_content).write\_pdf(stylesheets=[css\_styles])

**🔧 Implementation Priority**

**High Priority (Week 1)**

1. ✅ File structure reorganization
2. ✅ Template integration with enhanced design
3. ✅ CSS and JS asset organization
4. ✅ Basic multi-language support setup

**Medium Priority (Week 2)**

1. 🔄 Enhanced study plan generation logic
2. 🔄 Resource links integration
3. 🔄 Responsive design optimization
4. 🔄 Error handling and user feedback

**Low Priority (Week 3)**

1. ⏳ Advanced animations and interactions
2. ⏳ Analytics and progress tracking
3. ⏳ Performance optimization
4. ⏳ SEO enhancements

**📋 Required Dependencies Update**

# requirements.txt

Flask==2.3.3

Flask-Babel==3.1.0

Jinja2==3.1.2

python-dotenv==1.0.0

WeasyPrint==60.0

Pillow==10.0.0

gunicorn==21.2.0

**🚀 Deployment Considerations**

**Environment Variables**

# .env

SECRET\_KEY=your-secret-key-here

DEBUG=False

FLASK\_ENV=production

**Render Deployment**

# render.yaml

services:

- type: web

name: ielts-study-plan-generator

env: python

buildCommand: pip install -r requirements.txt

startCommand: gunicorn app:app

envVars:

- key: PYTHON\_VERSION

value: 3.11.0

**🔍 GitHub Repository Organization**

**Recommended Branch Strategy**

main # Production-ready code

├── develop # Development integration

├── feature/ui-enhance # UI enhancements

├── feature/backend # Backend improvements

└── hotfix/bugs # Critical bug fixes

**Missing Files to Add**

1. **Static Assets**: Logo files, icons, fonts
2. **Configuration**: Environment configs, deployment files
3. **Documentation**: API docs, user guide, changelog
4. **Testing**: Unit tests, integration tests
5. **CI/CD**: GitHub Actions workflows

**🎯 Next Steps**

1. **Backup Current Version**: Create a complete backup of your current working version
2. **GitHub Setup**: Ensure all current files are pushed to your repository
3. **Enhancement Integration**: Gradually integrate the enhanced HTML features
4. **Testing**: Test each enhancement thoroughly on both local and deployed versions
5. **Documentation**: Update README with new features and setup instructions

**📞 Support Resources**

* **GitHub Repository**: [Your repo link]
* **Live Demo**: https://ielts-app-1.onrender.com/
* **Documentation**: This comprehensive guide
* **Issue Tracking**: Use GitHub Issues for bug reports and feature requests

*This documentation serves as your complete roadmap for organizing, enhancing, and maintaining your IELTS Study Plan Generator application.*