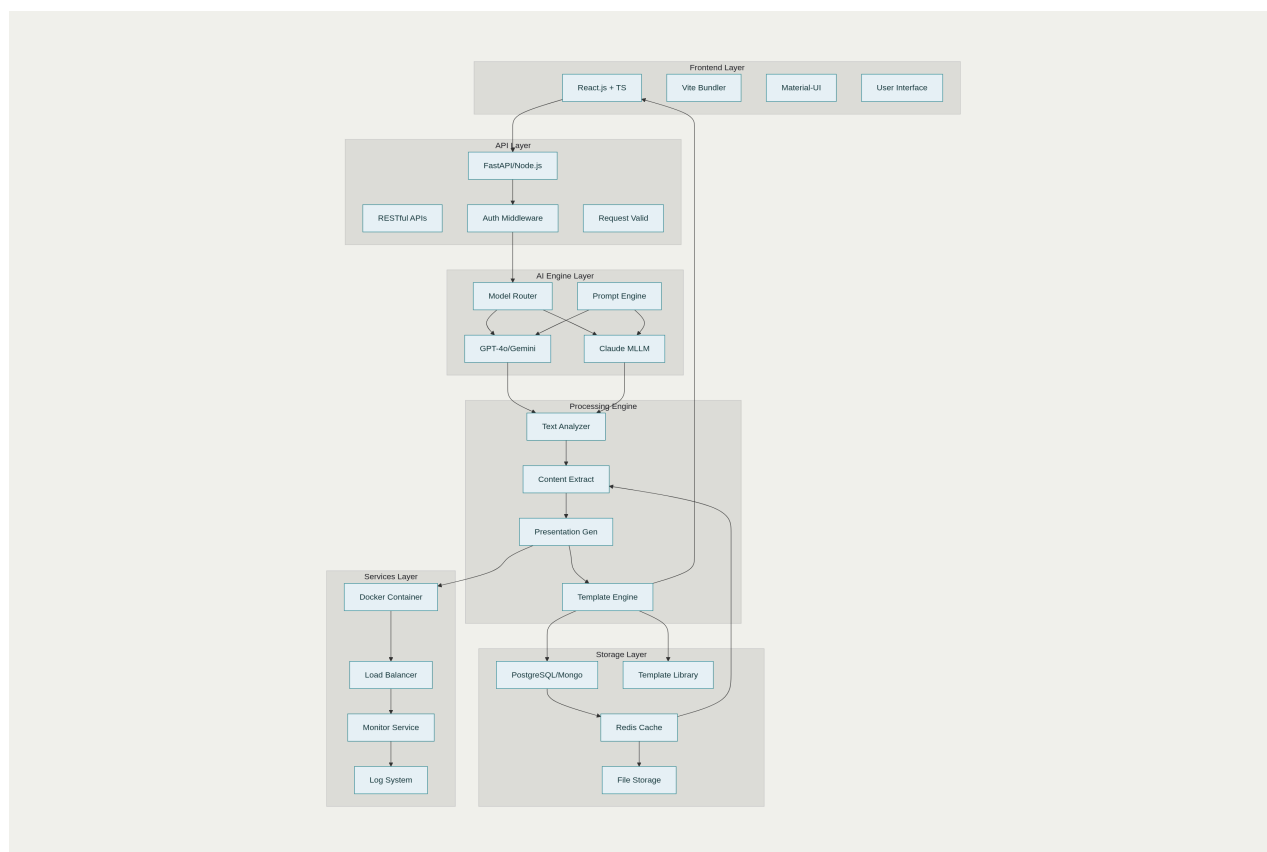




# 利用LLMs自動從文字產生簡報的Web AI Application 研究報告

## 執行摘要

基於大語言模型（LLMs）的自動簡報生成技術在2025年達到了前所未有的成熟度。最新研究顯示，**PPTAgent**框架以97.8%的成功率和3.67的整體品質評分領先業界，而**PreGenie**等多模態代理系統在視覺設計方面表現卓越。這些breakthrough技術結合先進的Web應用架構，為企業和個人用戶提供了高效、智能的簡報創作解決方案。<sup>[1] [2]</sup>



利用LLMs自動從文字產生簡報的Web AI Application技術架構圖

## 技術現狀與最新發展

## SOTA模型架構分析

2025年的簡報生成領域出現了三個主要技術範式的突破：

### 1. 編輯導向代理系統

PPTAgent採用革命性的兩階段編輯範式，首先分析參考簡報以理解結構模式和內容架構，然後通過程式碼行動來生成投影片。此方法避免了從零開始創建複雜簡報的挑戰，成功率從傳統方法的10%提升至97.8%。<sup>[1]</sup>

### 2. 多模態代理框架

PreGenie框架整合了五個專門的LLM和VLM，包括文字摘要器、圖像標註器、程式碼生成器等組件。該系統基於Slidev框架，支援Markdown格式的專業簡報生成，在設計品質方面獲得3.60分的高分。<sup>[2]</sup>

### 3. 直接整合系統

Talk-to-Your-Slides透過COM通訊直接操作PowerPoint物件，實現了真正的即時編輯功能。系統採用兩層架構：高層LLM代理解釋指令並制定編輯計劃，低層Python腳本直接操作PowerPoint對象。<sup>[1]</sup>

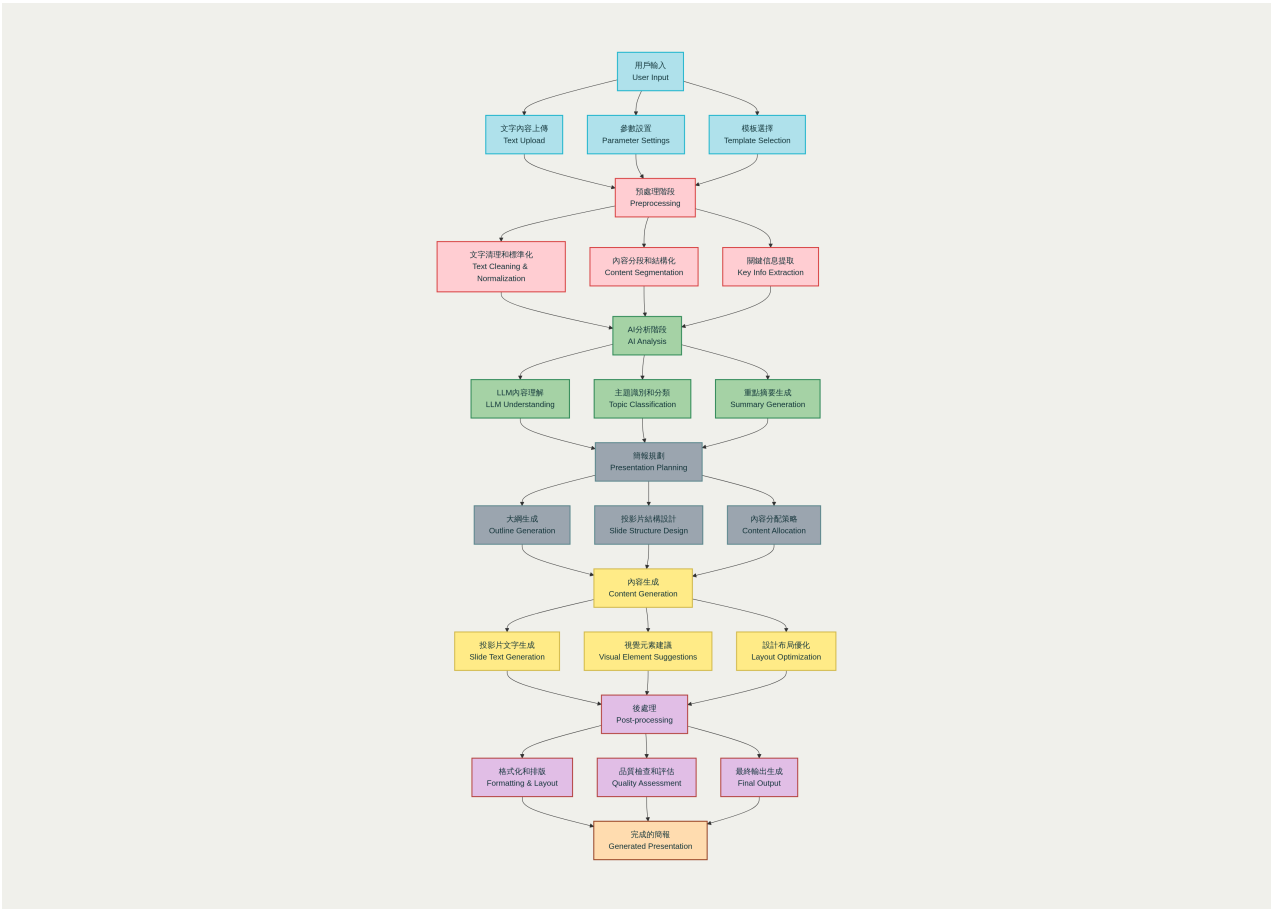
## 核心技術創新

### PPTEval評估框架

首個綜合性簡報評估系統PPTEval引入了三維評估標準：內容品質、視覺設計和邏輯連貫性。該框架採用MLLM-as-a-Judge範式，與人工評估的Pearson相關性達到0.71，顯著優於傳統ROUGE指標。<sup>[3]</sup>

### 多階段處理管線

最新的多階段LLM-VLM方法將簡報生成分解為內容提取、視覺整合和迭代優化三個階段。相比直接生成方法，這種approach在自動化指標和人工評估中都表現更優。<sup>[4] [5]</sup>



文字到簡報生成的AI處理流程圖

## Web應用架構設計

### 前端技術棧

**React 19.0 + TypeScript**作為主流選擇，提供並發渲染和Suspense特性。**Vite 6.1+**作為建構工具，支援熱模組替換和Tree-shaking優化。UI組件庫推薦**Material-UI v6**或**Chakra UI v3**，確保一致的用戶體驗。 [6] [7]

**狀態管理**採用**Zustand**或**React Query**處理複雜的資料同步需求。這些工具特別適合處理AI生成過程中的異步狀態更新和快取管理。 [7]

### 後端架構

**FastAPI 0.115+**因其異步處理能力和自動文檔生成功能成為首選。支援**WebSocket**連接以實現即時的生成進度反饋，以及**背景任務**處理耗時的AI推理過程。 [6] [7]

**資料庫設計**採用**PostgreSQL 16**存儲結構化資料，配合**Redis 7.2**提供毫秒級快取服務。檔案處理使用**python-pptx**庫進行PowerPoint格式操作。 [8] [9] [7] [10]

## API設計模式

**RESTful架構**結合**OpenAPI 3.1**規範確保類型安全和標準化。關鍵端點包括：<sup>[7]</sup>

- `POST /api/v1/presentation/generate` - 主要生成接口<sup>[11]</sup>
- `GET /api/v1/presentation/{id}/status` - 進度查詢<sup>[11]</sup>
- `PUT /api/v1/presentation/{id}/edit` - 即時編輯<sup>[11]</sup>

**請求參數**支援多種自定義選項：語調設定、詳細程度、投影片數量、模板選擇等。<sup>[11]</sup>

## 技術實現細節

### LLM整合策略

**多模型支援架構**允許開發者選擇適合的LLM提供者：OpenAI GPT-4o、Google Gemini、Anthropic Claude或本地Ollama模型。**模型路由器**根據任務複雜度自動選擇最適合的模型。<sup>[11]</sup>

**Prompt Engineering**採用結構化模板系統，針對不同簡報類型（商業、學術、教育）優化提示詞。**少樣本學習**技術提高特定領域的生成品質。<sup>[11] [12]</sup>

## 容器化部署

**Docker Compose**配置支援GPU加速和多服務編排。**Docker Offload**技術允許計算密集的AI推理任務無縫轉移到雲端GPU資源。<sup>[12] [13] [14]</sup>

```
services:
  api:
    build: ./backend
    ports: ["8000:80"]
    environment:
      - LLM=openai
      - OPENAI_API_KEY=${OPENAI_KEY}
  frontend:
    build: ./frontend
    ports: ["3000:80"]
```

## 效能最佳化

**快取策略**包括LLM回應快取、模板快取和圖像資源快取。**批次處理**能力允許同時處理多個簡報請求，提高資源利用效率。<sup>[11]</sup>

**CDN整合**確保全球用戶快速訪問生成的簡報檔案。**自動擴展**功能根據需求動態調整計算資源。<sup>[15]</sup>

## 評估與品質保證

## 評估指標體系

**PPTEval**框架建立了業界首個標準化評估體系：<sup>[3]</sup>

- **內容品質**：文字清晰度、資訊適當性、視覺支援度
- **設計品質**：色彩協調、視覺元素、整體美感
- **邏輯連貫性**：結構發展、背景資訊完整性

**TSBench**資料集提供379個多樣化編輯指令，用於系統性評估簡報編輯代理的能力。<sup>[1]</sup>

## 品質控制機制

**多層驗證**包括語法檢查、格式驗證和內容審核。**自我修正機制**允許系統偵測執行錯誤並提供即時回饋進行迭代改進。<sup>[1]</sup>

**A/B測試**框架支援不同模型和參數配置的效果比較。**使用者回饋整合**持續優化生成品質。<sup>[3]</sup>

## 商業應用與案例

### 企業級部署

**私有化部署**解決方案確保敏感資料不離開企業內部。**SSO整合**和**權限管理**支援企業級安全需求。<sup>[16]</sup>

**品牌客製化**功能自動應用企業視覺識別系統，保持品牌一致性。**模板庫管理**允許組織建立和分享標準化簡報模板。<sup>[16]</sup>

### 垂直應用領域

**教育科技**：自動課程簡報生成，支援多語言和個性化學習路徑<sup>[17] [18]</sup>

**商業諮詢**：快速提案簡報，整合資料視覺化和市場分析<sup>[19] [20]</sup>

**醫療健康**：病例簡報和研究發表，符合專業規範和隱私要求<sup>[17]</sup>

## 技術挑戰與解決方案

### 現有限制

**複雜佈局處理**：嵌套群組圖形的解析仍是瓶頸，影響結果一致性。**視覺一致性**：儘管有所改進，仍無法充分利用視覺提示來優化樣式一致性，可能出現元素重疊等設計缺陷。<sup>[1] [21]</sup>

**跨平台相容性**：COM通訊限制系統僅能在Windows環境運行，限制了使用者可及性。<sup>[1]</sup>

### 解決策略

**混合方法**：結合底層物件存取和UI視覺驗證，提高佈局品質。**跨平台API**：開發基於Web的PowerPoint API或平台無關的中介軟體解決方案。<sup>[1]</sup>

**視覺回饋整合**：納入渲染後的投影片圖像，實現更強健的視覺品質控制。<sup>[1]</sup>

## 未來發展趨勢

### 技術演進方向

**多模態融合**：整合語音、視頻和互動元素的下一代簡報系統。**即時協作**：支援多人同時編輯和AI輔助的智能協作平台。<sup>[22]</sup>

**個性化引擎**：基於使用者偏好和歷史資料的高度個性化內容生成。**跨語言支援**：原生多語言生成和即時翻譯功能。<sup>[23] [24]</sup>

### 產業標準化

**開放格式支援**：擴展至更多簡報格式，包括Web標準和新興格式。**API標準化**：建立業界統一的簡報生成API規範。<sup>[11] [25]</sup>

**倫理AI整合**：確保生成內容的準確性、公平性和透明度。<sup>[1]</sup>

### 技術建議與最佳實踐

#### 開發建議

1. **採用微服務架構**：分離AI推理、檔案處理和用戶介面服務，提高系統可維護性<sup>[7]</sup>
2. **實施快取策略**：多層快取設計減少LLM API呼叫成本和延遲<sup>[11]</sup>
3. **建立監控系統**：全面的效能監控和錯誤追蹤確保服務穩定性<sup>[15]</sup>

#### 部署策略

1. **容器化優先**：使用Docker和Kubernetes實現一致的開發和生產環境<sup>[12] [13]</sup>
2. **雲原生設計**：支援多雲部署和自動擴展能力<sup>[15] [13]</sup>
3. **安全防護**：實施API限流、身份驗證和資料加密<sup>[16]</sup>

### 結論

利用LLMs自動從文字產生簡報的Web AI應用已進入成熟期，以PPTAgent和PreGenie為代表的SOTA系統展現了令人印象深刻的技術能力。透過採用編輯導向範式、多模態代理架構和綜合評估框架，這些系統在成功率、內容品質和視覺設計方面都取得了顯著突破。

Web應用架構的最佳實踐結合React、FastAPI、Docker和雲原生技術，為開發者提供了構建可擴展、高效能簡報生成服務的完整解決方案。隨著技術持續演進，預計將看到更多創新應用和商業模式的出現，推動整個產業向更智能、更人性化的方向發展。

對於希望進入此領域的開發者和企業，建議採用開源框架如Presenton作為起點，結合最新的LLM技術和Web開發最佳實踐，構建符合特定需求的定制化解決方案。關鍵成功因素包括正確的技術架構選擇、穩健的品質評估機制，以及持續的用戶體驗優化。<sup>[11]</sup>



2. <https://arxiv.org/html/2505.21660v2>
3. <https://arxiv.org/html/2501.03936v3>
4. <https://aclanthology.org/2024.inlg-main.18/>
5. <https://arxiv.org/abs/2406.06556>
6. <https://dev.to/stamigos/modern-full-stack-setup-fastapi-reactjs-vite-mui-with-typescript-2mef>
7. <https://testdriven.io/blog/fastapi-react/>
8. <https://towardsdatascience.com/how-to-use-llms-to-create-presentation-slides-genslide-a-step-by-step-guide-31f7588ffb5e/>
9. <https://dev.to/composiodev/tired-of-making-ppt-slides-see-how-i-automated-it-in-under-50-lines-of-code-4gbo>
10. <https://github.com/fastapi/full-stack-fastapi-template>
11. <https://github.com/presenton/presenton>
12. <https://www.c-sharpcorner.com/article/build-and-deploy-ai-apps-easily-with-docker-compose/>
13. <https://www.docker.com/solutions/docker-ai/>
14. <https://www.docker.com/blog/build-ai-agents-with-docker-compose/>
15. <https://www.mulesoft.com/platform/deploy-api>
16. <https://presenti.ai/private-deployment/>
17. <https://arxiv.org/abs/2501.06497>
18. <https://www.slidesai.io>
19. <https://slidesgo.com/ai/presentation-maker>
20. <https://www.presentations.ai/tools/text-to-ppt>
21. <https://aiscientist.substack.com/p/musing-101-pptagent-generating-and>
22. <https://neptune.ai/blog/multimodal-large-language-models>
23. <https://www.canva.com/create/ai-presentations/>
24. <https://www.linkedin.com/pulse/evaluation-metrics-generative-ai-systems-vinay-ananth-zw8uc>
25. <https://www.flashdocs.com/post/the-best-apis-to-create-powerpoint-presentations-in-2025>
26. <https://arxiv.org/html/2501.03936v1>
27. <https://slidemodel.com/text-to-powerpoint-ai/>
28. <https://www.youtube.com/watch?v=RRC3mE4emlo>
29. <https://www.adobe.com/express/create/ai/presentation>
30. [https://www.linkedin.com/posts/seldo\\_long-weekend-so-an-unusually-complicated-activity-7236533028313800705-RMAF](https://www.linkedin.com/posts/seldo_long-weekend-so-an-unusually-complicated-activity-7236533028313800705-RMAF)
31. <https://plusai.com/blog/best-ai-presentation-makers>
32. <https://slidespeak.co>
33. <https://dev.to/copilotkit/how-to-build-an-ai-powered-powerpoint-generator-langchain-copilotkit-openai-nextjs-4c76>
34. <https://github.com/icip-cas/PPTAgent>
35. [https://www.reddit.com/r/ChatGPTPromptGenius/comments/1g20606/best\\_llm\\_module\\_for\\_creating\\_presentations/](https://www.reddit.com/r/ChatGPTPromptGenius/comments/1g20606/best_llm_module_for_creating_presentations/)
36. <https://www.aitoolssme.com/comparison/ai-tools-for-presentations>

37. <https://www.beautiful.ai>
38. <https://hyper.ai/en/headlines/1f992fa6cfb7b2649a611c841b2c8f83>
39. <https://www.mentimeter.com/features/ai-builder>
40. <https://twistly.ai>
41. <https://openreview.net/pdf/037ab5d3d72e7037b5a6ff7a9a9b786ca8488b1c.pdf>
42. [https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=8991&context=etd\\_theses](https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=8991&context=etd_theses)
43. <https://www.confident-ai.com/blog/llm-evaluation-metrics-everything-you-need-for-llm-evaluation>
44. <https://www.dataknobs.com/generativeai/10-llms/>
45. <https://slidesgpt.com>
46. <https://encord.com/blog/generative-ai-metrics/>
47. [https://www.linkedin.com/posts/kkuhl\\_marp-editor-activity-7325922237247557635-xUhZ](https://www.linkedin.com/posts/kkuhl_marp-editor-activity-7325922237247557635-xUhZ)
48. <https://www.presentations.ai>
49. <https://cloud.google.com/vertex-ai/generative-ai/docs/models/determine-eval>
50. <https://www.tomkytran.com/how-to-use-gpt-4-to-create-an-entire-powerpoint-presentation/>
51. <https://blog.google/technology/ai/generative-media-models-io-2025/>
52. <https://tilburg.ai/2024/05/how-to-use-chatgpt-4o/>
53. <https://hiringnet.com/image-generation-state-of-the-art-open-source-ai-models-in-2025>
54. [https://www.reddit.com/r/LLMDevs/comments/1e9yr5x/revolutionizing\\_slide\\_creation\\_new\\_llm\\_vlm\\_hybrid/](https://www.reddit.com/r/LLMDevs/comments/1e9yr5x/revolutionizing_slide_creation_new_llm_vlm_hybrid/)
55. [https://www.linkedin.com/posts/nicoleleffer\\_if-youre-just-using-chatgpt-deep-research-activity-7316614786878746624-RmgX](https://www.linkedin.com/posts/nicoleleffer_if-youre-just-using-chatgpt-deep-research-activity-7316614786878746624-RmgX)
56. <https://www.getadigital.com/blog/the-current-state-of-ai-image-generation-as-of-early-2025>
57. <https://aclanthology.org/2024.inlg-main.18.pdf>
58. <https://www.stateof.ai>
59. <https://towardsdatascience.com/multimodal-models-llms-that-can-see-and-hear-5c6737c981d3/>
60. [https://www.reddit.com/r/OpenAI/comments/1ec5m57/can\\_gpt\\_4o\\_or\\_any\\_other\\_llm\\_output\\_powerpoint/](https://www.reddit.com/r/OpenAI/comments/1ec5m57/can_gpt_4o_or_any_other_llm_output_powerpoint/)
61. <https://icml.cc/virtual/2025/poster/45841>
62. <https://blog.langchain.com/multi-modal-rag-template/>
63. <https://www.youtube.com/watch?v=xNlBlseLPkM>
64. <https://www.sciencedirect.com/science/article/pii/S2468502X24000160>
65. <https://dev.to/apilover/how-to-start-building-apis-in-2025-a-comprehensive-guide-58k5>
66. <https://slidespeak.co/guides/top-5-ai-presentation-apis-2025>
67. <https://powerdrill.ai/discover/summary-pptagent-generating-and-evaluating-presentations-cm5ptd7de88ci07sftowyb28a>
68. <https://www.cmarix.com/blog/web-api-development/>
69. <https://www.themoonlight.io/en/review/pptagent-generating-and-evaluating-presentations-beyond-text-to-slides>
70. <https://devico.io/blog/complete-guide-to-api-development-in-2025>
71. <https://openreview.net/pdf/f9bd183085916baafdf6a045b868fa75c1d8200f.pdf>



72. <https://arxiv.org/pdf/2503.17710.pdf>
73. <https://budibase.com/blog/web-application-development/>
74. <https://github.com/aws-samples/generate-your-presentation-with-llm>
75. <https://trangotech.com/blog/web-application-development-trends/>
76. <https://www.youtube.com/watch?v=qGL35HMkj4M>
77. <https://docs.crawl4ai.com/core/docker-deployment/>
78. <https://christophergs.com/tutorials/ultimate-fastapi-tutorial-pt-12-react-js-frontend/>
79. <https://learntocloud.guide/phase3/deploy-api>
80. <https://www.slideshare.net/slideshow/cloud-deployments-done-right-why-apis-are-key/63575958>
81. <https://www.docker.com/blog/docker-for-web-developers/>
82. [https://www.youtube.com/watch?v=\\_1P0Uqk50Ps](https://www.youtube.com/watch?v=_1P0Uqk50Ps)
83. <https://cloud.google.com/apis>
84. <https://docs.docker.com/compose/gettingstarted/>
85. [https://www.reddit.com/r/FastAPI/comments/1h0kcd6/fastapi\\_react\\_full\\_stack/](https://www.reddit.com/r/FastAPI/comments/1h0kcd6/fastapi_react_full_stack/)
86. <https://www.infoq.com/presentations/api-design-implement-document/>
87. <https://dev.to/oayanda/docker-compose-deploy-a-containerized-php-application-52ll>
88. <https://fastapi.tiangolo.com/project-generation/>
89. <https://ppl-ai-code-interpreter-files.s3.amazonaws.com/web/direct-files/fe60cb96559bb5b6f1faea53a0b3cecc/135cc244-be98-456e-a70c-681fbb469c11/8239571f.csv>
90. <https://ppl-ai-code-interpreter-files.s3.amazonaws.com/web/direct-files/fe60cb96559bb5b6f1faea53a0b3cecc/59d9d646-d92a-4dd2-a5ed-86b5341affa5/46406652.csv>
91. <https://ppl-ai-code-interpreter-files.s3.amazonaws.com/web/direct-files/fe60cb96559bb5b6f1faea53a0b3cecc/fe44d46c-1466-46e8-8d6a-c86b8f64bd31/5ae8425b.csv>