1. Project Overview

AI-Powered Email Campaign Builder for Personalized Marketing

Problem Statement:
Small business owners, creators, and marketers often struggle with creating email campaigns that are both professional and personalized. Writing emails, formatting them for design, and planning a sequence of messages all take time and expertise that not everyone has. Current email marketing tools do not offer strong personalization or intelligent content generation based on the specific content a business wants to promote.

Objectives and Goals:

- Build an AI-powered web application that automatically creates email campaigns based on a website URL or uploaded document.
- Use natural language processing to understand the source content and generate email subject lines, body text, and calls to action.
- Enable mass personalization of emails using contact data such as names and user behavior.
- Make the system simple enough for non-technical users to create and send campaigns quickly.
- Ensure the system is fully built and deployed by the development team with no third-party AI services used for core logic.

2. Scope and Use Case

Target Audience:

- Small business owners who want to promote products or services through email.
- Content creators, freelancers, and solopreneurs who want to stay connected with their audience.
- Marketing beginners who need help generating content and running campaigns.

Use Case Scenarios:

- A business owner pastes their website URL into the app. The system reads their homepage and generates a 3-part email sequence promoting their product.
- A creator uploads a PDF eBook. The app creates follow-up emails encouraging readers to book a call or buy something related.

- A user uploads a CSV of contacts and selects a campaign goal. The system sends personalized emails to each contact with their name and behavior tags.

Project Constraints:

- Resources: Must use free or low-cost hosting tools.
- Team: Two-person development team with beginner to intermediate skill level.

3. Data Requirements

Data Sources:

- Data comes from URLs or uploaded PDFs provided by users.
- Public datasets with email templates and marketing copy styles used to train the AI components.
- Examples of email campaign language sourced from blogs, open-source email sets, and simulated marketing texts.

Data Preprocessing Needs:

- Clean website text or PDF content to remove menus, ads, and unrelated sections.
- Use tokenization to split content into useful pieces (sentences, paragraphs).
- Extract keywords, product descriptions, tone, and audience style.
- Organize training data to include examples of subject lines, CTAs, and email sequences.

Data Labeling:

- Training data is organized by type (e.g., subject line, greeting, CTA). Some labels are applied manually or generated by scripts during setup.
- Personalization variables (e.g., name, product type) are marked for later dynamic content generation.

4. Model Selection

Model Type:

- A fine-tuned transformer model will be developed for generating marketing email content.
- For content extraction, lightweight rule-based models and scraping tools will be used (BeautifulSoup, PyMuPDF).

Model Justification:

- Transformer-based models are good at understanding and generating natural language, making them ideal for creating high-quality emails.
- We chose to build and fine-tune our own small transformer model rather than using an API to ensure the project meets the build-from-scratch requirement.

Transfer Learning:

- A base transformer model will be fine-tuned using our cleaned and labeled email dataset.
- Libraries such as Hugging Face Transformers will be used in training, with our own training process and datasets.

5. Project Timeline and Milestones

Timeline:

- **Week 1:** Define scope, collect datasets, begin basic frontend and backend structure.
- **Week 2:** Implement content scraping and parsing from URLs and PDFs.
- **Week 3:** Train and test transformer model for generating subject lines and email content.
- **Week 4:** Build the backend system with FastAPI to connect model outputs to user actions.
- **Week 5:** Develop frontend interfaces for campaign building, contact management, and previewing emails.
- **Week 6:** Testing, bug fixes, deployment, and collection of user feedback.

Milestones:

- **Week 2:** Content extraction module working with input from URLs and PDFs.
- **Week 3:** Model generates emails with subject lines and body content based on input.
- **Week 5:** Frontend and backend connected for full campaign flow from input to email preview.
- **Week 6:** System is deployed and working end-to-end with personalization and scheduling.

6. Evaluation and Testing Plan

Testing Environment:

- Local testing using test URLs, PDFs, and fake contact lists.

- Full-stack integration tested in development mode on Azure.
- Final deployment tested in production on Azure App Services.

User Feedback Collection:

- App shared with classmates, instructors, and small business contacts.
- Simple feedback form included in the app for users to report issues or confusion.

Error Handling and Monitoring:

- Model prediction errors and failed email deliveries logged in Supabase.
- Alerts and logs reviewed weekly to fix bugs and improve accuracy.
- In future, error logs can be used to retrain and improve the model.

7. Ethical and Responsible AI Considerations

Data Privacy:

- No user content is stored without permission. Content is processed in memory only.
- Emails and contact data stored in encrypted format on Supabase.

Fairness and Bias:

- Model trained on diverse examples from different industries to avoid bias.
- Outputs reviewed manually during development to check for inappropriate or offensive content.
- Users can edit all generated content before sending.

Environmental Impact:

- Using pre-trained models and small datasets to reduce energy use.
- Training done on local machines or small cloud instances with efficient settings.

8. Project Outcomes and Impact

Expected Impact:

- Help small teams and individuals save time and create more professional email campaigns.
- Offer a new way to personalize emails at scale, increasing engagement and conversion.

- Allow non-marketers to compete with larger teams by using AI to automate writing and design tasks.

Future Extensions:

- Add multilingual support for international users.
- Train separate models for different industries like education, health, or real estate.
- Integrate A/B testing features to compare different email versions.
- Allow saving of custom templates and campaign histories.

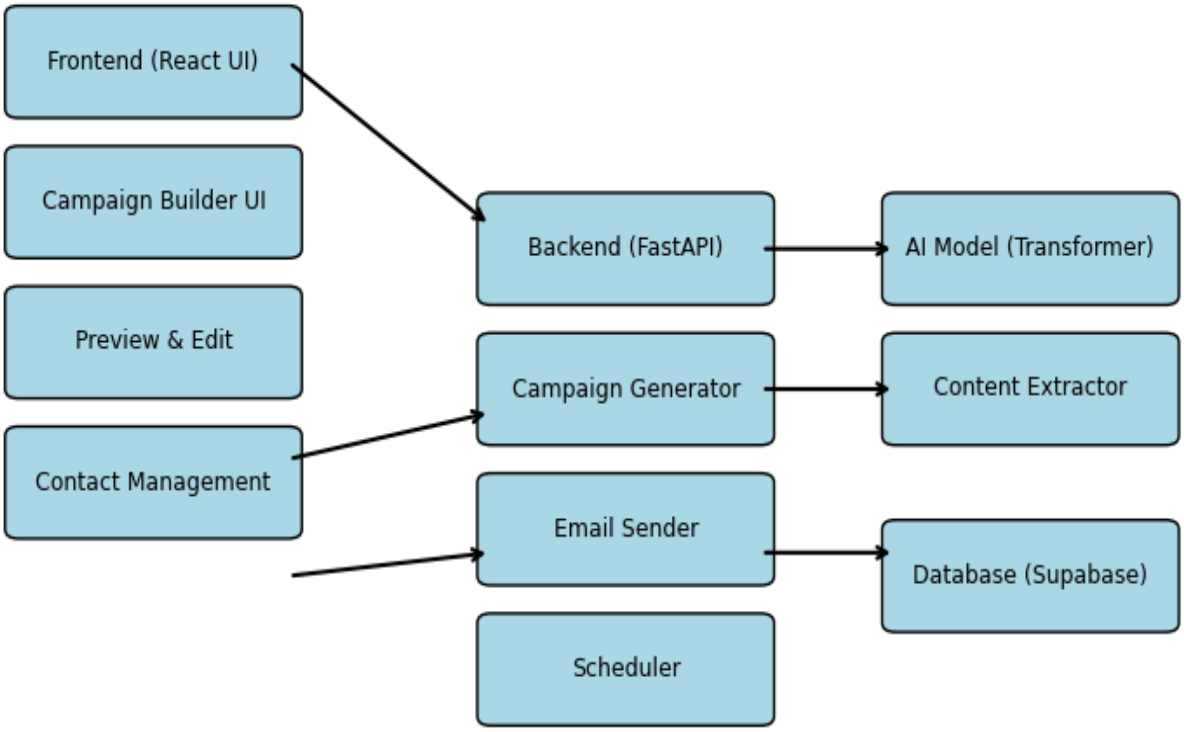
9. References

Resources and Research:

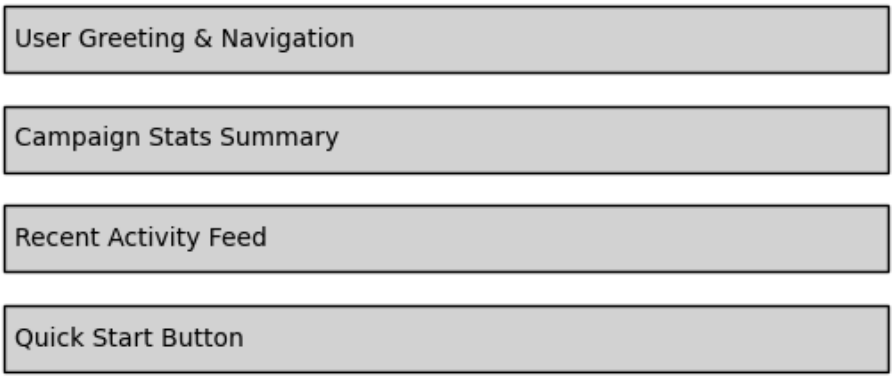
- Hugging Face Transformers Documentation: https://huggingface.co/transformers/
- PyMuPDF Documentation: https://pymupdf.readthedocs.io/
- BeautifulSoup Web Scraping Guide: https://www.crummy.com/software/BeautifulSoup/
- OpenAI Transformer Papers and Tutorials (for baseline understanding)
- Blogs and articles on email marketing strategy (HubSpot, Mailchimp, etc.)
- Open datasets from marketing and copywriting examples on GitHub and Kaggle

Data Sources:

- Simulated email campaign datasets
- Public web content used under fair use
- User-uploaded materials provided during testing and evaluation


Frontend (React UI)
Campaign Builder UI
Preview & Edit
Contact Management
Backend (FastAPI)
Campaign Generator
Email Sender
Scheduler
AI Model (Transformer)
Content Extractor
Database (Supabase)

Dashboard UI Mockup



Campaign Builder UI Mockup

Input URL or Upload PDF

Select Campaign Goal (Sales, Newsletter, Follow-up)

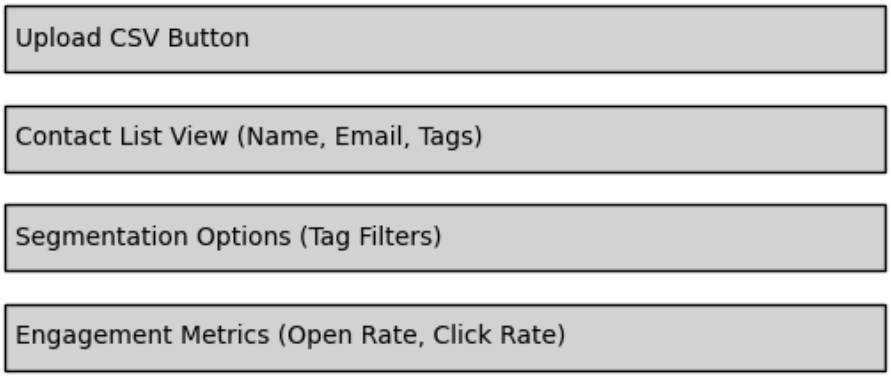
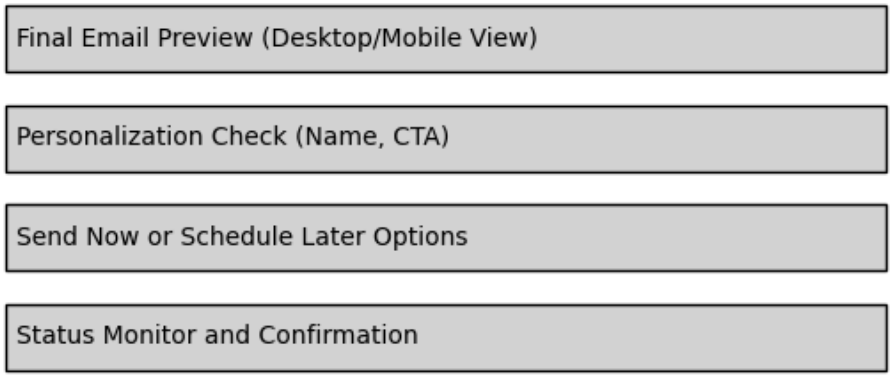
AI-generated Content Preview

Edit Subject Line and Email Body

CTA Suggestions and Image Placeholders

Save and Continue Button

Contact Manager UI Mockup



Email Preview & Scheduling UI Mockup