



YOUR DAILY BREAD

SENIOR PROJECT PROPOSAL

AUTHOR INFORMATION

Allie Peterson

alpeterson@westmont.edu

Westmont College

PREPARED FOR

CS-195: Computer Science Senior Seminar

Prof. Mike Ryu

dongyub.ryu@gmail.com

Westmont College

CREDITS/ACKNOWLEDGEMENTS

Mike Ryu, Phil Tanaka, Dan Shank: Product Vision and Design discussion

All of the material included in this course

Henry Matisse: Floral graphic squares



PROJECT DESCRIPTION

Project Overview

This product is a web-based platform that uses generative AI and engages users in a quiz-like format to create personalized Christian devotionals for each user. Upon visiting the website, users are prompted with questions relating to their cultural context and their spiritual curiosities and struggles and are prompted to choose a response from a multiple-choice lineup. These responses are then processed AI models that generate tailored, biblically accurate, and culturally relevant devotional for the user that speaks to their unique struggles and questions.

The product aims to provide an intimate spiritual experience, using technology to provide the user with tools that open them up to Biblical resources and content. This product utilises the advantages of generative AI in a faith setting without the moral and relational dangers that come with the chatbot experience.

Motivation

As the use and popularity of generative AI grows rapidly, it is crucial to implement this tool in a way that doesn't cross ethical boundaries. With this being such a powerful tool, completely rejecting it in religious and Church contexts is foolish and unrealistic. In creating a product that interacts with or provides spiritual experiences, the designer must be careful not to provide a product that can replace crucial aspects of faith such as fellowship, mentorship, and community with the body of believers. In a chatbot model, the conversational experience inherently develops an emotional connection between the user and the agent—especially when the conversation surrounds the intensely vulnerable topics of sin, repentance, love, and others. An application that provides devotionals avoids this as users read devotionals as content and interact with them as inputting information opposed to a spiritual mentor or evangelist. Even though the code and backend of these two products could be quite similar, the expectation and relationship the user has with the

product is crucial in protecting fellowship among believers.

Problem Statement

The problem this project aims to solve is the overwhelming nature of the bible, where it can be confusing to know where to start or find a passage that speaks to a certain struggle. Additionally, the Bible was written to a first-century Greco-Roman context, which can make it challenging to translate its messages into its readers' modern cultural context.

Proposed Solution

To address the challenge of navigating the vastness of biblical content and the complexity of its cultural context, this product will guide users with prompts to help them clarify what they want to explore. It will then generate a biblically-based devotional tailored to communicate those themes through their own cultural lens. This solution guarantees users receive relevant biblical content that speaks to their questions and curiosities in a way that resonates with their unique cultural context.

Specifications

Frontend (React)

The frontend will be developed in React in order to create a dynamic and interactive web application. I may use additional tools to manage quiz inputs (React Hook Form or other) and UI libraries to make the feel professional and clean (Material UI or other).

Backend (Python)

The backend will be implemented in Python along with frameworks to support user requests. For the generative AI piece, Langchain and OpenAI's API will be used to process quiz responses and generate curated devotionals.

Datasets

In order to ensure the text-generated content is biblically-sound, the model will pull from datasets not limited to, but likely including Open Bible API, various open source biblical commentaries, Desiring God, and BibleGateway. For the cultural representation, the model will pull from linguistic datasets like Open Multilingual Wordnet and cultural contextual resources including but not limited to Enthologue and other data sets holding cultural data.

Deployment (AWS)

For deployment, the frontend will be hosted on AWS services and the backend with AWS Lambda, along with GitHub Actions integration for CI/CD automation.

Justifications

Novelty

While there are a few bible-based chatbots out there (including apologist.ai and askjesuschatbot.com), they are not tailored towards cultural context or acknowledge that many biblical questions don't have a single answer, making this product a unique and valuable asset to the market. This product also takes a unique approach in its user interaction taking place through quiz response selections instead of a chatbot interface.

Feasibility

With the thorough documentation and popularity of the primary tools used for this project (Python, React, LangChain, OpenAI's APIs, AWS services), the resources are plentiful for the combination and application of these technologies. Given the familiarity of these tools and the scope of the project, completion of an accurate and functioning biblical-based text generator at the end of April is feasible.

Cost

The majority of the tools that will be used for this project are free, with AWS and openAI as exceptions. Hosting a website on AWS at this scale will cost between \$1 and \$3 a

month, which is quite reasonable. With \$15 worth of credit already purchased for the OpenAI API and considering the active weekly users will likely stay below 30 after release, the cost for initial usage will be sufficiently covered.



PROJECT GOALS

Mission Statement

Bringing biblical daily bread to all who are hungry, made with the unique ingredients of their own cultural context and questioning.

OKRs

Tasks		Grading Criteria
1 - Alpha Launch		
Objective	Build a functioning prototype of the AI devotional generator that generates biblically accurate devotional texts best on user quiz choices.	
Key Result 1	Product API is set up using OpenAI and Langchain to handle quiz input and generate devotional text, with the API responding correctly to a minimum of 70% of the test queries.	A- API responds to 70% of test queries.
		B- API responds to 60-69% of test queries.
		C- API responds to 50-59% of test queries.
		D- API responds to 40-49% of test queries.
		F- API responds to under 40% of test queries.

Key Result 2	The AI model generates a minimum of 3 coherent, biblically-accurate devotionals that pass content quality checks by 3 theological Reviewers (Shawn Peterson, Rahcel Koch, Westmont Staff).	A- AI model generates 3 devotionals approved by theological reviewers.
		B- AI model generates 2 devotionals approved by theological reviewers.
		C- AI model generates 1 devotional approved by theological reviewers.
		D- AI model generates devotionals that fail theological review.
		F- AI model does not generate devotionals.
Key Result 3	Frontend is established using React adhering to Alpha-launch wireframes for UI/UX.	A- Frontend matches wireframes and includes more detail.
		B- Frontend is representative of wireframes almost identically.
		C- Frontend is representative of wireframes but is sparse and simple.
		D- Frontend is extremely basic and lacks relation to wireframes.
		F- Frontend is blank and has not been implemented.

2 - Beta Launch	
Objective	Refine the devotional generator to deliver personalized, culturally relevant, and user-friendly devotionals by addressing feedback from beta testing.

Key Result 1	Diversification of devotional content is achieved by incorporating data reflecting 3 cultural contexts: U.S., Korea, and Mexico.	A- Each of the 3 cultural contexts is represented.
		B- $\frac{2}{3}$ of the cultural contexts are represented.
		C- $\frac{1}{3}$ of the cultural contexts are represented.
		D- Some cultural contexts are represented, but not the ones listed.
		F- No cultural contexts are represented.
Key Result 2	Beta testing with 10 users provides feedback on the accuracy, relevance, and ease of use, ensuring an average satisfaction score of 75% on satisfaction surveys.	A- 10 users produce a 75% satisfaction score.
		B- 7-9 users produce a 60-74% satisfaction score.
		C- 5-7 users produce a 50-59% satisfaction score.
		D- Under 5 users produce a satisfaction score under 50%.
		F- No user feedback.
Key Result 3	70% of the critical issues from beta testing are addressed and resolved.	A- 70% issues are addressed and resolved.
		B- 60-69% issues are addressed and resolved.

		C- 50-59% issues are addressed and resolved.
		D- 40-49% issues are addressed and resolved.
		F- Under 40% issues are addressed and resolved.

3 - Final Launch		
Objective	Bring people closer to Christ by providing a personalized and biblically-rooted devotional generator that is globally relevant, user friendly, and widely impactful.	
Key Result 1	Data from 5 countries, 5 bible topics, and 5 personal struggles (minimum) are used to generate devotionals.	A- Minimum of 4 countries, 5 bible topics, 5 personal struggles included.
		B- Exactly 3 countries included.
		C- Less than 3 countries included.
		D- Biblical topics or personal struggles dip below 2.
		F- One of the 3 is missing: countries, bible topics, or personal struggles.
Key Result 2	A bug-free experience is achieved with fewer than 5 critical bugs reported at the time of release.	A- Fewer than 5 bugs at release.
		B- More than 5 bugs at release.
		C- More than 7 bugs at release.
		D- More than 8 bugs at release.

		F- More than 9 bugs at release.
Key Result 3	Within the first 2 weeks of deployment, the site hits 10 active users, with satisfaction survey results achieving an average satisfaction score of 80% or higher.	A- Site has 10 active users with 80% satisfaction.
		B- Site has 8-10 active users with 70-79% satisfaction.
		C- Site has 6-8 active users with 80% satisfaction.
		D- Site has 6-8 active users with 50-79% satisfaction.
		F- Site has less than 6 active users or a satisfaction score of less than 50%.