

Team-Project-18 -> Project Ideas

Project Idea 1. Codeless Application Development

Introduction to the problem statement:

In a world where Industry is moving towards Agile model, rapid development and delivery of working software products is a huge challenge. Codeless Application Development platform intends to reduce the gap between requirements and the deliverable product by using Natural Language processing and Deep Learning.

Abstract (Rough draft):

Codeless Application Development platform takes in user requirements in Well-formed English Language sentences and can generate the code for required functionality in required programming languages such as Python, JavaScript, Java and SQL etc. This platform then also provides an option to push that function/code generated by NLP onto a repository, which is linked to a containerized CI/CD pipeline that can create/update an application instance on the cloud that's deliverable instantly. The user has an option to either push the code or just verify that the code generated by the platform is in sync with their requirements which were input to the system. The platform also can support tasks such as generating English text which explains the given input program/function code.

Approach

The platform uses OpenAI's GPT-3 API under the hood to take in Requirements in simple English sentences and to generate logical and accurate code. Generative Pre-trained Transformer 3 (GPT-3) is an autoregressive language model that uses deep learning to produce human-like text. GPT-3's deep learning neural network is a model with over 175 billion machine learning parameters.

This platform intends to use the excellent text generation capabilities of GPT-3 to generate codes in different programming languages by passing requirements as prompt to the API call. Once the response is generated, the platform would be able to display the code to a user, save the particular requirement and response in the user's Code generation history and will give the user an option to push the code to repositories such as GitHub. The platform will be able to link and authenticate user's GitHub profile securely to enable pushing code. We can either choose to write our own build files for CI/CD pipelines or try to generate them from the platform's code generator. Codeless Application Development platform enables code generation from requirements, renders it to the user to verify and provides options to push the code into containerized pipelines to turn it into a deliverable piece of software.

Persona

- Businesses, Companies, Developers on the lookout for Rapid Development and Deliverables.
- Non-Techies who want to build things without deep diving into the tech infrastructure.
- Students who want to understand the flow of Agile Development

Links/References:

OpenAI GPT3: <https://openai.com/blog/gpt-3-apps/>

Serverless computing options to package and deliver the generated code:

IBM: <https://developer.ibm.com/depmodels/serverless/>

AWS: <https://aws.amazon.com/lambda/>

RedHat: <https://cloud.redhat.com/learn/topics/serverless>

Project Idea 2. JARVIS - A personal assistant portal

Introduction to the problem statement:

With all the multiple things one is involved in daily - Academics, Personal technical growth, Job prep, blogging, meetings, socializing, fitness, self-improvement etc, we plan to create a single portal (a web application) wherein users can keep track of their progress across each dimension.

Abstract (Rough draft):

Logging Component: - This will be the one base component that will be available throughout. - This component will accept a text or image input at a certain point in time. And once the user saves the data, it will be persisted in a database (mostly MongoDB) under appropriate title. - Why image input? - Because at times, we usually take notes physically with pen and paper. Once done, user can have it clicked and just upload it on the portal

Progress Component: - This will be the one base component that will be available for each Module. - It will be present alongwith each component and it will render the number of hours spent day wise. eg.

```
-Sept 1 - 2 hrs
-Sept 2 - 0 hrs
-Sept 3 - 0 hrs
-Sept 4 - 0 hrs
-Sept 5 - 0 hrs
-Sept 6 - 0 hrs
-Sept 7 - 0 hrs
-Sept 8 - 4 hrs
-Sept 9 - 0 hrs
-Sept 10 - 1 hrs
-Sept 11 - 0 hrs
-Sept 12 - 0 hrs
-Sept 13 - 0 hrs
-Sept 14 - 2 hrs
-Sept 15 - 0 hrs
-Sept 16 - 0 hrs
-Sept 17 - 3 hrs
-Sept 18 - 0 hrs
-Sept 19 - 0 hrs
-Sept 20 - 0 hrs
-Sept 21 - 0 hrs
-Sept 22 - 5 hrs
-Sept 23 - 0 hrs
-Sept 24 - 6 hrs
-Sept 25 - 0 hrs
-Sept 27 - 0 hrs
-Sept 28 - 0 hrs
-Sept 29 - 1 hrs
-Sept 30 - 0 hrs
```

We can then give metrics like:

- users longest continuous streak
- plot a graph of users hours spent

MODULES:

1. Academics
 - (Lecture notes etc.)
 - (I feel this would be very useful if later, say after a year or two, I quickly remember that some professor had mentioned something in his lecture and I want to refer to that...)
 - To ensure that a student gives equal amount of time to all his courses.
2. Personal technical growth
 - Personal projects,
 - Technologies to learn (it usually happens you learn something of some tech, but then since you didn't log it, you don't recollect the progress you had done earlier; leading you to start all over or worst – drop the learning all together)
 - Certifications, exams prep
3. Job Prep
 - Internship search
 - Leetcode
3. Blogging
 - Say you solved a technical problem after hours of googling and you want to record it step-by-step.
4. Meetings
 - To store all the MoMs (Minutes of Meeting)
5. Personal Space
 - Entertainment
 - Collate all your music playlists (links or youtube video embeds) at one place like youtube music, spotify, Amazon Music etc.
 - Motivation / quotes etc.
 - you can have your own articles, or you can make some notes over some motivational video that you saw on youtube.

Approach

- The application will be a MERN application.
- Since ReactJS is used for frontend, we may use React-Native to build an app for it

Persona

- Students

Project Idea 3. Audio Sentiment Analysis

Introduction to the problem statement:

One of the major issues faced by Customer Care call centres is that customers rarely give feedback if they are satisfied with the service or not. As such, these centres rarely have enough data to improve on.

Abstract (Rough draft):

Every call recorded by the call centre can be fed to a program that will analyse the tone and in turn show how satisfied the customer was. This saves time on behalf of the customer as well as these agencies. Most of the work is around text based sentimental analysis. This project suggests to take it one step further and analyse audio.

Approach

A front-end to upload the voice samples and backend and running a server that will analyse and return the output, which will be stored in a database. The basic approach in analysing will be – detect human speech using VAD, recognize the speaker, speech and finally analyse the sentiment.

Persona

- Call centres, customer care, etc.

Links/References:

<https://heartbeat.comet.ml/sentiment-analysis-of-speech-using-pydup-and-speechrecognition-in-python-8a0f3bdabc80>

<https://cloud.google.com/architecture/visualize-speech-data-with-framework>

<https://arxiv.org/ftp/arxiv/papers/1802/1802.06209.pdf>

Project Idea 4. Voice based chat bot

Introduction to the problem statement:

If You want to provide better service over the phone, but often limited resource planning simply does not allow that ideal level of improvement. This is a common challenge. When it comes to automation, many people seem to have reservations about the quality of customer service. But the opposite is the case: Why should it be bad that a VoiceBot is used for questions that can be answered automatically and therefore quickly and at any time? If you give people the choice of either waiting longer to be able to speak to a person, or offering a response directly from a VoiceBot, many will opt for artificial intelligence.

Abstract (Rough draft):

Voice Bots(VBs) have been a trending topic in recent years. As of the last decade, VBs have emerged with the aim of simplifying human-machine interactions and found a wide use case in the market. For example, Siri and Google Assistant are some of the most well-known VB's developed by the tech giants Apple and Google. Whether navigating the web or messaging on a phone, it is likely that VBs have been confronted offering the user help. In this code pattern we will create a Machine Learning VB, but the twist here is that we'll be using voice input and output. For the conversation dialog we'll be using VoiceBot Assistant, but we'll also be using Voicebot Speech To Text to capture the user's voice, and lastly we'll use Voicebot's Text To Speech to playback the VoiceBots response to the user.

Approach

A customer begins a spoken or digital interaction using natural, conversational language. The bot uses natural language processing (NLP) to understand the customer's intent and find the right solution. The VoiceBot here recognizes it and processes the customer's request and replies to the customer in speech. Use the integrated Natural Language Understanding (NLU) menu items to automatically apply machine learning to understand customer intents.

Persona

-VoiceBots in Sales influences customers' decision-making process by handle customers' emotions well, making them more likely to purchase the products -VoiceBots in Education make it easier for a student to be informed about the critical events in their student life such as learning the deadlines for assignments and their final grades on a course -Home automation, which is achieved by integrating IoT devices with voice assistants, is considered one of the main uses of voice assistants -Can be especially helpful for visually impaired people or people who have limited mobility

Links and references

Google's Phone Gateway: <https://cloud.google.com/dialogflow/es/docs/integrations/phone-gateway>

IBM Watson's AI solution: <https://www.ibm.com/cloud/ai/customer-service>