Cohere AI Thanksgiving

Vacancy finder

Lablab.ai

Diana Kiikbayeva

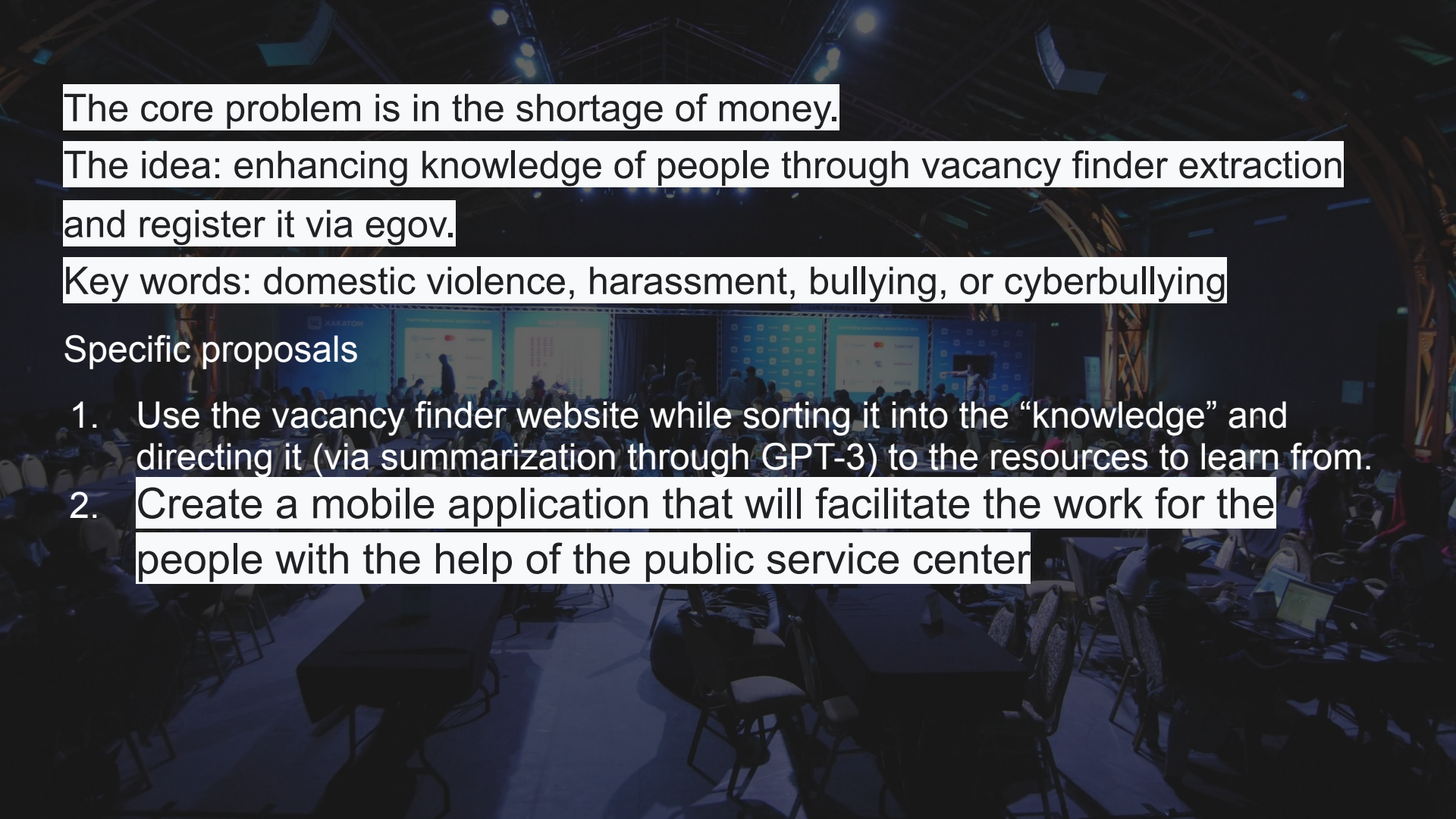
OpenAI

new native group

NEXTGRID

lab lab ai

In Kazakhstan, the problem of violence in society remains relevant. According to statistics, every fifth woman is forced to put up with domestic violence. Every year, about 400 women die from domestic violence in the country. There is also the problem of harassment, which is faced by one in four women and who faces violations of their rights in the workplace, at school, as well as in public transport.



The core problem is in the shortage of money.

The idea: enhancing knowledge of people through vacancy finder extraction and register it via egov.

Key words: domestic violence, harassment, bullying, or cyberbullying

Specific proposals

1. Use the vacancy finder website while sorting it into the “knowledge” and directing it (via summarization through GPT-3) to the resources to learn from.
2. Create a mobile application that will facilitate the work for the people with the help of the public service center

```
[ ] from google.colab import files
    from IPython.display import Image
```

```
[ ] uploaded = files.upload()
```

Выбрать файлы Файл не выбран Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving imagecode.png to imagecode.png
```

```
[ ] Image('imagecode.png', width =800);
```

```
[ ] pip install easyocr openai matplotlib python-dotenv
```

```
[ ] import os
    import cv2
    import openai
    import easyocr
    import matplotlib.pyplot as plt
    from torch.cuda import is_available
    from dotenv import load_dotenv
```

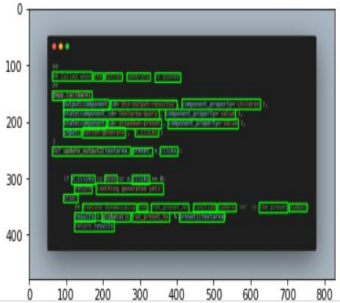
```
[ ] class Reader:
        def __init__(self, is_cuda=False):
            self.reader = easyocr.Reader(['en'], gpu=is_cuda, model_storage_directory=os.path.join('models'), download_enabled=True)

        def __call__(self, img):
            return self.extract_text(img)

        def extract_text(self, img, show_text=False, show_confidence=False):
            result = self.reader.readtext(img)

            extracted_text = []

            for text in filter(lambda x: x[-1] > .45, result):
                box, acc_text, confidence = text

                # box[0] and box[2] - upper left and lower right corners of the box
                img = cv2.rectangle(img, [int(i) for i in box[0]], [int(i) for i in box[2]], (0, 255, 0), 2) # each coordinate is a list has to be int

                if show_text and show_confidence:
                    img_text = f'{acc_text} - ({"{:.3f}".format(confidence)}%)'
```

```
[ ]
            elif show_text:
                img_text = acc_text

            elif show_confidence:
                img_text = f'CONF: ({"{:.3f}".format(confidence)}%)'

            if show_text or show_confidence:
                img = cv2.putText(
                    img,
                    img_text,
                    (int(box[0][0]), int(box[0][1] - 3)),
                    cv2.FONT_HERSHEY_SIMPLEX,
                    fontScale=.5,
                    color=(168, 90, 50),
                    thickness=2
                )

            extracted_text.append(acc_text)

        return extracted_text, img
```

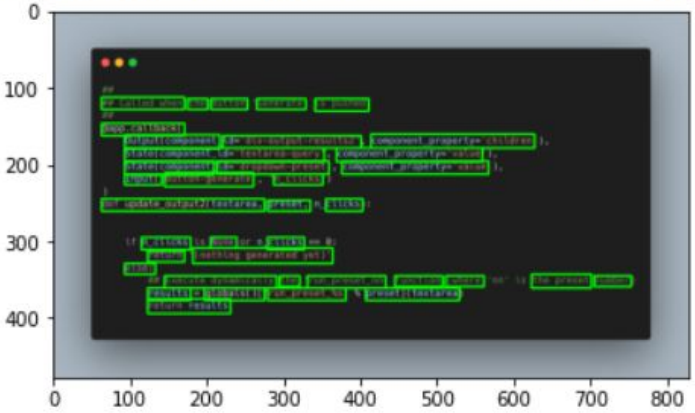
```
[ ]  def read_img(img_path):
        img = cv2.imread(img_path)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        return img
    if __name__ == '__main__':

        reader = Reader(is_cuda=is_available())
```

```
[ ]
    img = read_img('imagecode.png')
    text, extracted_image = reader(img)

    text = ' '.join(text)

    print('Extracted_text')
    print(text)

    plt.imshow(extracted_image)
    plt.show()
```

```
WARNING:easyocr.easyocr:Using CPU. Note: This module is much faster with a GPU.
WARNING:easyocr.easyocr:Downloading detection model, please wait. This may take several minutes depending upon your network connection.
Progress: |                                        | 100.0% CompleteWARNING:easyocr.easyocr:Downloading recognition model, please w
Progress: |                                        | 100.0% CompleteExtracted_text
## Called when the Button Generate is pushed @app. callback( Output(component id-'div-output-resultsz' component_property-"children State( co
```



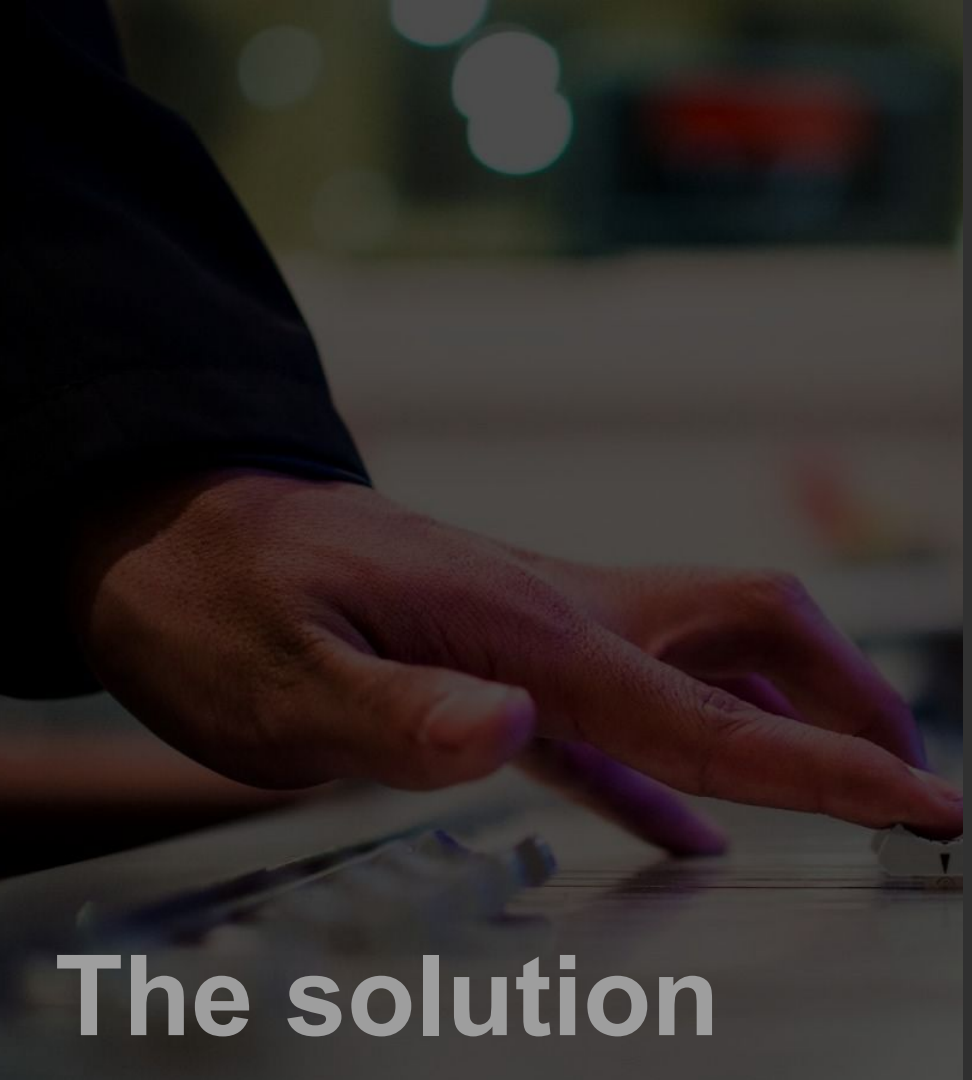
```
WARNING:easyocr.easyocr:Using CPU. Note: This module is much faster with a GPU.
WARNING:easyocr.easyocr:Downloading detection model, please wait. This may take several minutes
Progress: |                                                                                | 100.0% CompleteWARNING:easyocr.e
Progress: |                                                                                | 100.0% CompleteExtracted_text
## Called when the Button Generate is pushed @app. callback( Output(component id-'div-output-re
```



```
[ ]
```

```
text = ' '.join(text)

print('Extracted_text')
print(text)

plt.imshow(extracted_image)
plt.show()
```


How my product can
solve the problem
and how it works



The solution

- Extract the image of certain vacancy (job) website
- Get the extracted text from the image together
- Register the vcacy through public service center using an Adalo app.

untitled5.ipynb

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка

Поделиться

+ Код + Текст

Подключиться

Редактирование

```
from google.colab import files
from IPython.display import Image
```

```
[ ] uploaded = files.upload()
```

```
[ ] Image('imagecode.png', width =800);
```

```
[ ] pip install easyocr openai matplotlib python-dotenv
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting easyocr
  Downloading easyocr-1.6.2-py3-none-any.whl (2.9 MB)
     |████████████████████████████████| 2.9 MB 7.4 MB/s
Collecting openai
  Downloading openai-0.25.0.tar.gz (44 kB)
     |████████████████████████████████| 44 kB 2.9 MB/s
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
    Preparing wheel metadata ... done
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (3.2.2)
Collecting python-dotenv
  Downloading python_dotenv-0.21.0-py3-none-any.whl (18 kB)
Collecting opencv-python-headless<=4.5.4.60
  Downloading opencv_python_headless-4.5.4.60-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (47.6 MB)
```

It is important to register vacancy at the public service center. Hence, the app has been created using non-code.

Database Collections

Database Collections

Users 1 Record

Queue 28 Records

Name

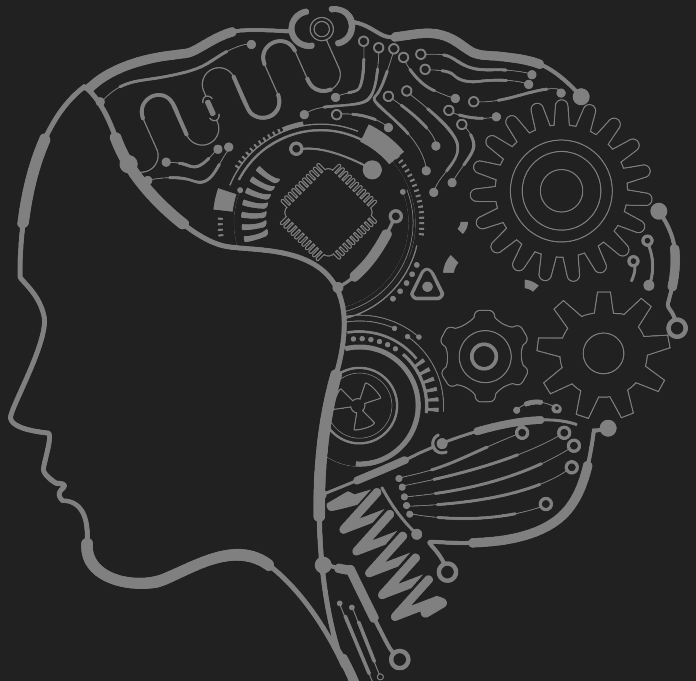
номер очереди

рабочее время

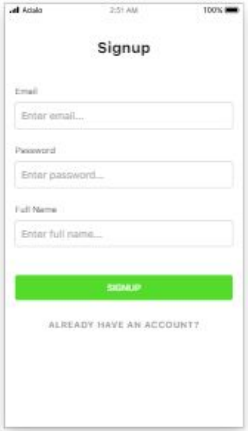
Users

ADD PROPERTY

ADD COLLECTION

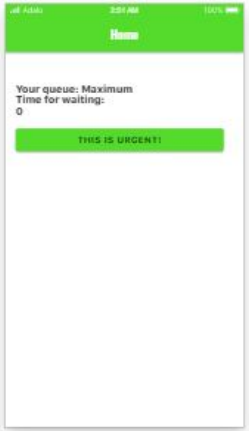


Signup

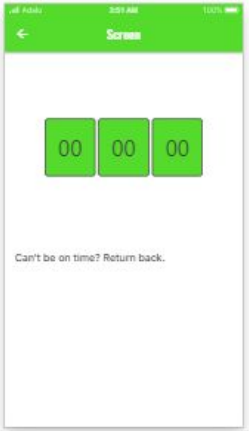


Login

Home

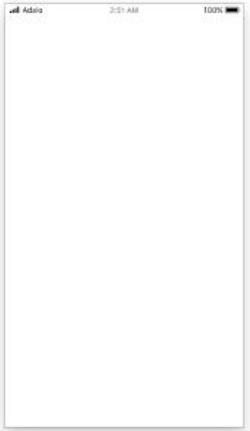


Screen

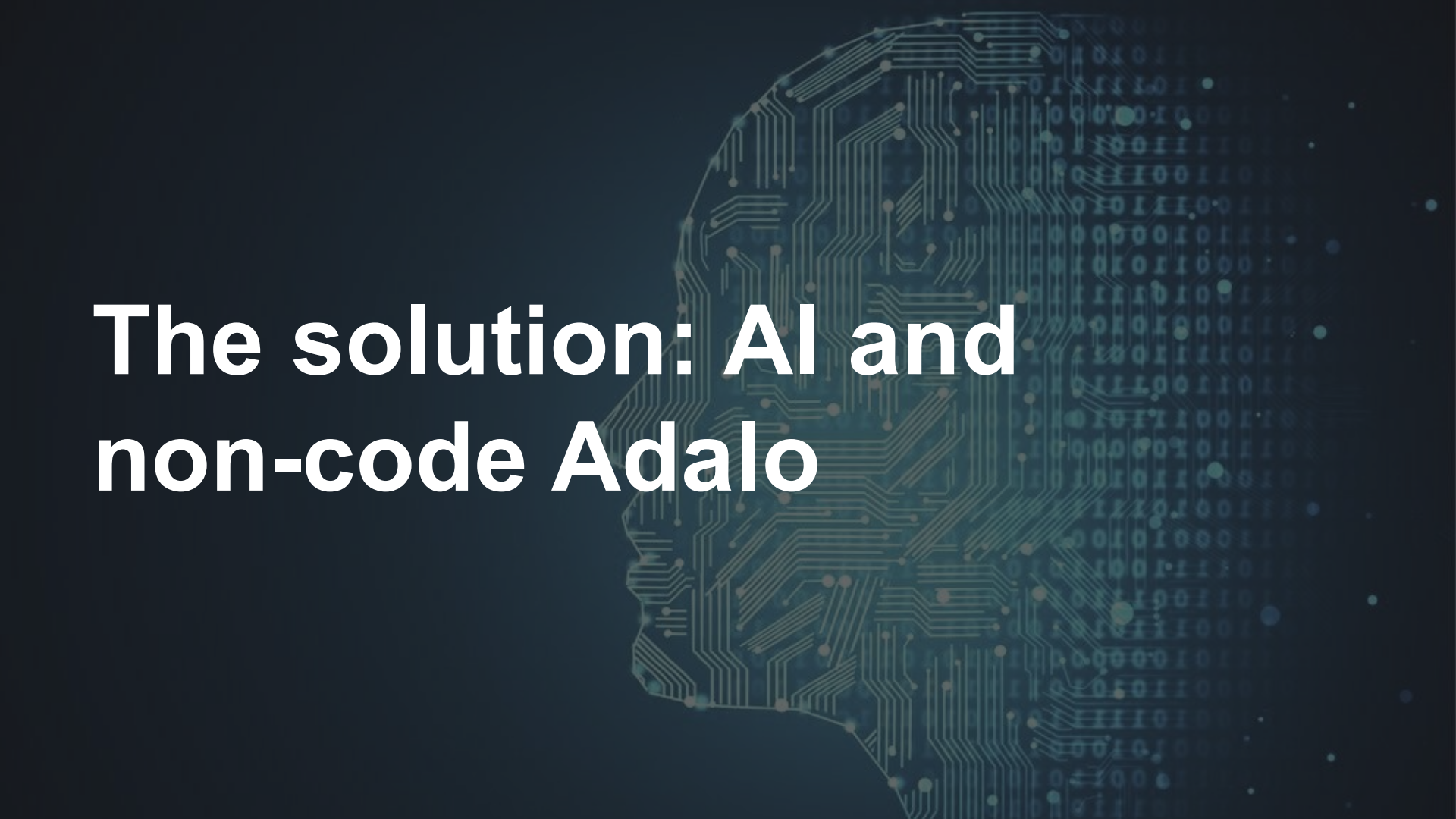


Screen 2

Screen 3



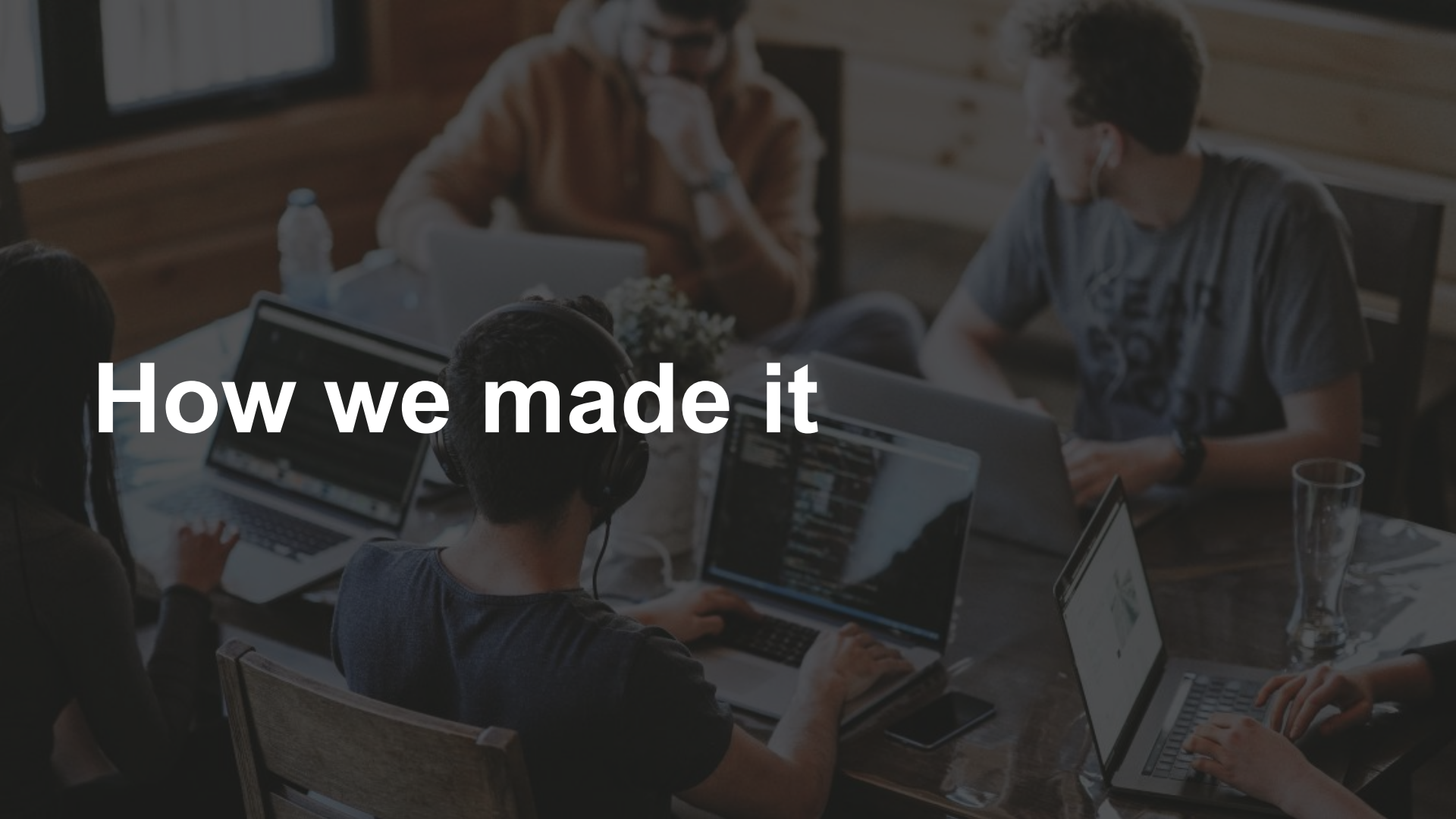
The solution: AI and non-code Adalo



A working demo of it

The links are:

1) https://colab.research.google.com/drive/102yEZro5k8yEg1z6rEKoNsdKu819ThBL#scrollTo=YX6-mq-_P3UX
2) https://previewer.adalo.com/760441d5-d40c-4bf2-88b7-562256584866https://previewer.adalo.com/760441d5-d40c-4bf2-88b7-562256584866



How we made it

For this, a screen recording of how you perform a common user's interaction might be perfect

XSplit

Link

OBS Studio

Link

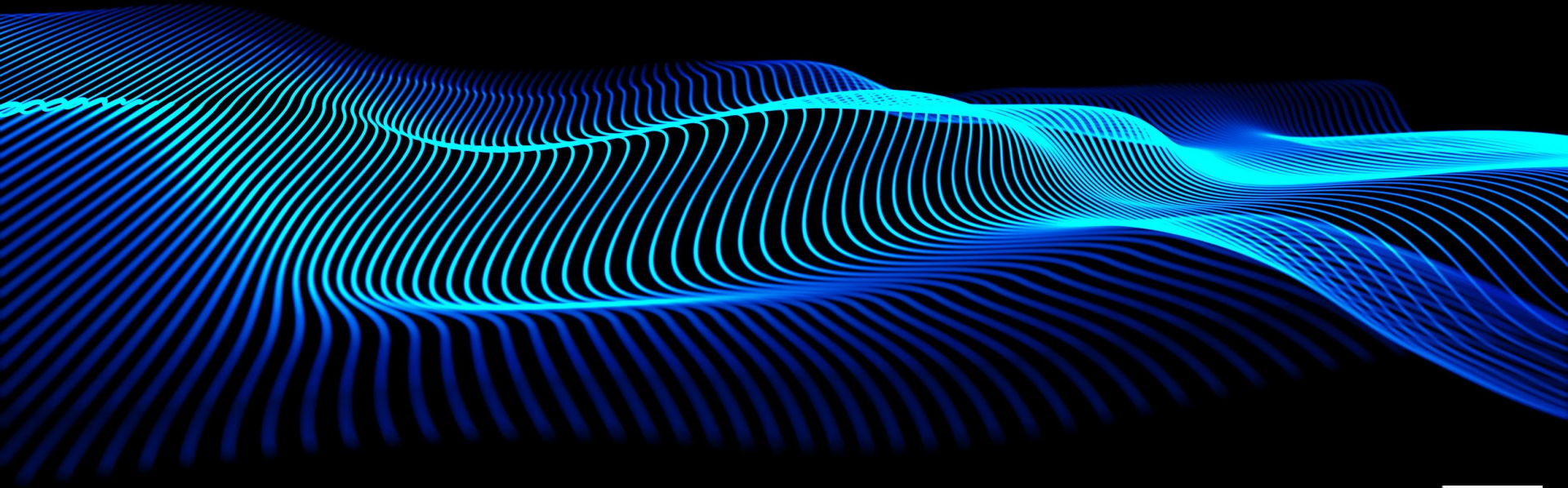
Screencast-O-Matic

Link

MONOSNAP

Link

Thanks!



OpenAI

new native group

NEXTGRID

lab lab ai