MIT – DESIGNING AND BUILDING AI PRODUCTS AND SERVICES – TAKE AWAYS

Patrick Stingley (CovertChannel@yahoo.com), Lynne Williamson (DiaKahina@aol.com), Daniel Wilson (dan@infoscience.ai), Miriam Ayala (MiriamNoe93@gmail.com)

ABSTRACT

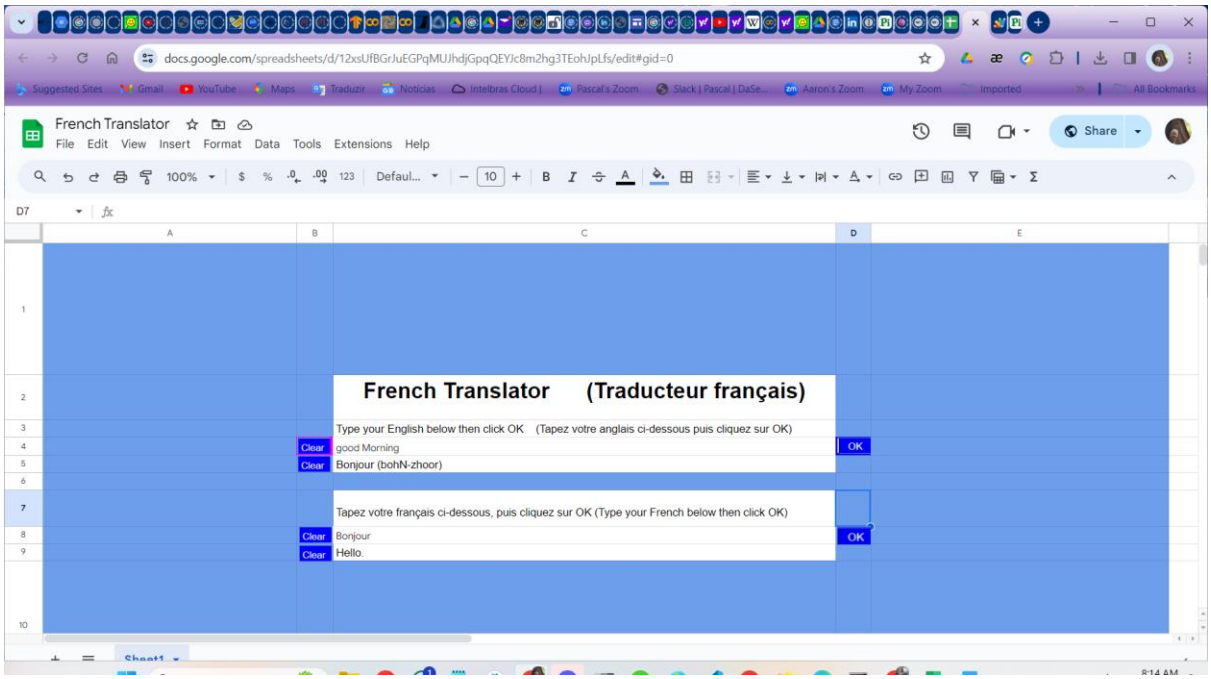
A few students in the Designing and Building AI Products and Services developed the following solutions so that all of the students of this course could take an AI application and process documentation with them to help them in their future endeavors.

This paper describes a prototype. It is not intended to be a working product, but instead is a working prototype to demonstrate how AI systems can leverage Large Language Models (LLMS) to provide valuable services. In this case, the service delivered is language translation with transliteration.

BACKGROUND AND REQUIREMENTS

The goal was to provide a working AI project that any student could take with them, so extremely low cost was a primary driver. It is possible to create a zero cost web page by using Google Sheets, so this platform was selected. The URL for the demo is:

https://docs.google.com/spreadsheets/d/12xsUfBGrJuEGPqMUJhdjGpqQEYJc8m2hg3TEohJpLfs/edit#gid=0 This was shortened to: bit.ly/3QhiJrH



Although this website is open to anybody who has access to it, because it's Google Sheets, they will have to log in to their Google account to use it.

PROCEDURES

To use the site follow these steps:

1. Click the clear buttons to clear the data entry cells.
2. Type the English text the user wants translated in cell C4.
3. Click in another cell such as D2 to release focus from the text entry cell.
4. Click the OK button in cell D4

   The translated words and a transliteration to provide pronunciation guidance should appear in cell C5

To translate the other language into English, put the other language into cell C8 and follow the above steps.

TECHNOLOGY

This prototype is an example of how many new businesses will be created in the next couple of years. It implements the Façade design pattern, using a web page (in this case delivered by Google Sheets) to a façade to present a specific function to the user. On the back end, the user's input is wrapped in a prompt and sent to OpenAI's ChatGPT. The ChatGPT then does the translation and transliteration and sends the response back to Google Sheets. Google Sheets then posts the response in cells C5 (or C9)

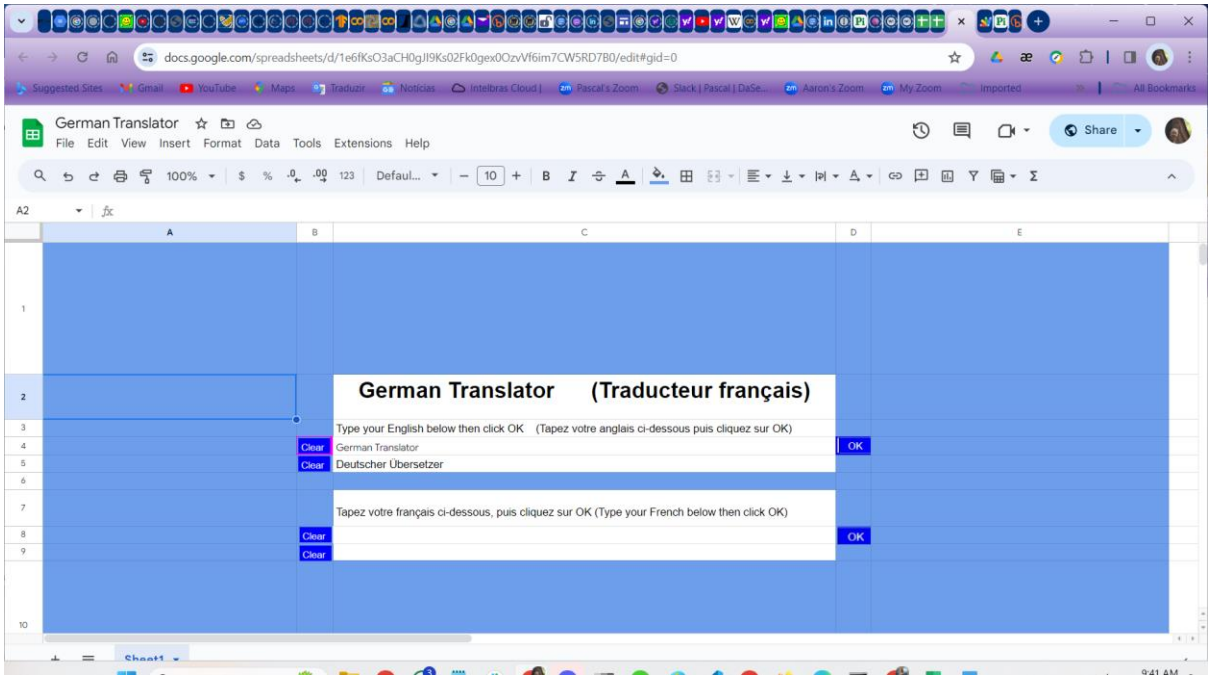
To use this, the user must have a paid account with OpenAI. We're sorry, but we were not able to get this to work with the free version of ChatGPT.

The code behind Google Sheets is Java Script, which none of us had any experience in, so we got help from ChatGPT. If the code looks primitive, that's OK. We kept it as simple as possible so people could customize it as they saw fit.

The first thing one would want to do, would be to go into File -> Make a Copy to make their own copy of this page in Google Sheets. Then follow these steps:

1. Change the title to whatever use they may want. In this case, the copy was named German Translator.
2. Next, choose the Extensions menu item in Google Sheets. Then select Apps Script. This is where the code behind the sheet is located. (The code is shown in Appendix 1.)
3. Replace `YOUR KEY GOES HERE` with your own key. This should be in between the single quotes on the first line of this code.
4. To turn this into a German translation site, replace the word French with German on line 7.
5. Click the Deploy button
6. Select "new deployment"
7. Add a description if desired. It is not necessary.
8. Click Deploy
9. Click Done
10. Go back to the spreadsheet and test it. (Often, it is best to re-load the page.)
11. It will now be necessary to give Google Sheets permission to run this script (click OK)
12. Select your Google account
13. Click Advanced.

14. Go to Untitled project (unsafe)
15. Click Allow



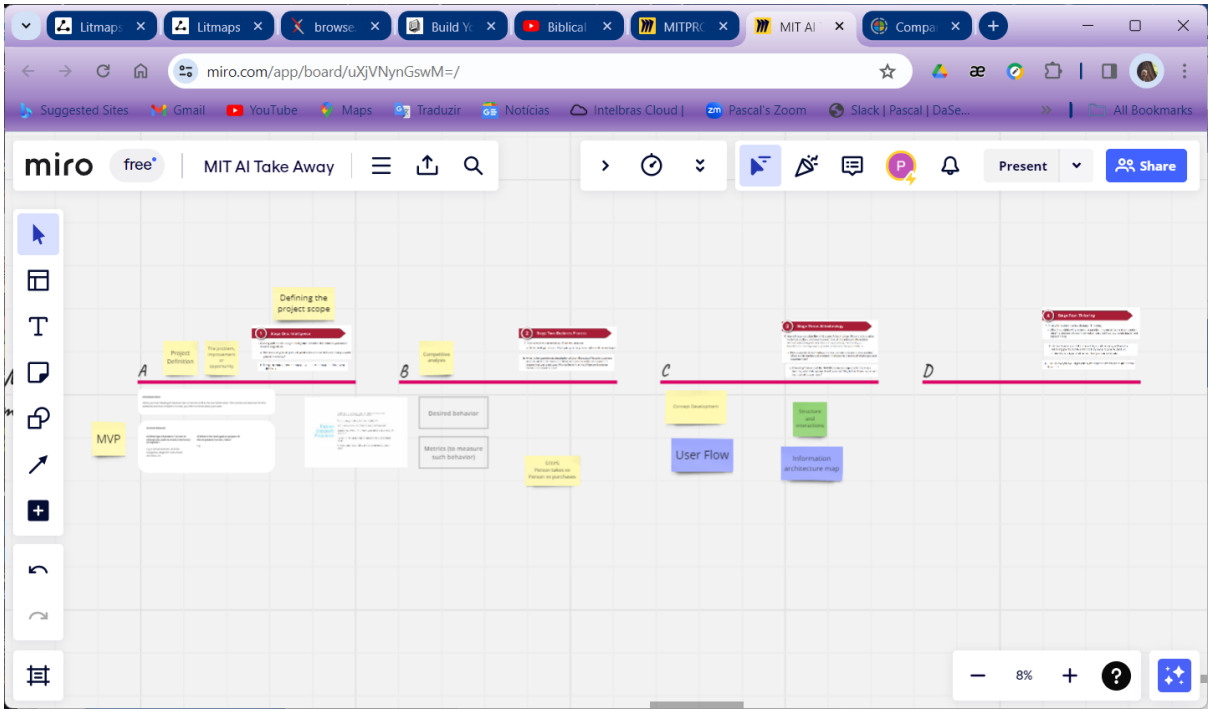
The system should now work. So, in this case, the tool is used to translate the words for the Google sheet as shown. In this case, Traducteur Francais will be replaced by Deutscher Übersetzer.

Feel free to customize this as you see fit. This same template has been used to get jokes from ChatGPT and other things.

PROJECT PLANNING

The team used miro (miro.com) to perform the accompanying planning aspects of this project. There is a free option, so one person started the board and then invited others to join.

Each person contributed to the master Miro board, and then used the components they felt were helpful on their own project. We found that this collaborative approach yielded better results than working independently.



CONCLUSION

In conclusion, it is hoped that all students, of this cohort, as well as subsequent ones will find this helpful in their studies and in life afterward.

Appendix 1.

```
const OPENAI_API_KEY = 'YOUR KEY GOES HERE'; // ChatGPT API key

function sendEnglishToChatGPT() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var promptCell = sheet.getRange("C4"); // Assuming the prompt is in
cell C4
  var promptText = promptCell.getValue();
  var fullPrompt = 'Translate the following into French with
transliteration: '+promptText;
  var response = callOpenAI(fullPrompt);
  var responseCell = sheet.getRange("C5"); // The response will be
written in cell C5
  responseCell.setValue(response);
}

function sendOtherLanguageToChatGPT() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var promptCell = sheet.getRange("C8"); // Assuming the prompt is in
cell C8
  var promptText = promptCell.getValue();
  var fullPrompt = 'Translate the following into English: '+promptText;
  var response = callOpenAI(fullPrompt);
  var responseCell = sheet.getRange("C9"); // The response will be
written in cell C9
  responseCell.setValue(response);
}
function clearC4(){
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var clearC4 = sheet.getRange("C4");
  clearC4.setValue("");
}

function clearC5(){
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var clearC4 = sheet.getRange("C5");
  clearC4.setValue("");
}

function clearC8(){
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var clearC4 = sheet.getRange("C8");
  clearC4.setValue("");
}

function clearC9(){
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var clearC4 = sheet.getRange("C9");
  clearC4.setValue("");
}
```

```
function callOpenAI(fullPrompt) {
  var url = 'https://api.openai.com/v1/chat/completions';
  var payload = {
    "model": "gpt-3.5-turbo",
    "messages": [{"role": "system", "content": "You are a helpful
assistant."}, {"role": "user", "content": fullPrompt}]
  };

  var options = {
    'method' : 'post',
    'contentType': 'application/json',
    'headers': {
      'Authorization': 'Bearer ' + OPENAI_API_KEY
    },
    'payload' : JSON.stringify(payload)
  };

  try {
    var response = UrlFetchApp.fetch(url, options);
    var jsonResponse = JSON.parse(response.getContentText());
    return jsonResponse.choices[0].message.content.trim();
  } catch (e) {
    Logger.log('Error: ' + e.toString());
    return "Error in fetching response";
  }
}
```