Course name: csc 131-02

Sprint# 1

Team name: ENIAC

Team members:
Ayush Thapa, Bilal Baloch, Anthony Torres, Dylan Khon

Sprint start date: 09-19-23
Sprint end date: 10-03-23

Sprint tasks and goals (be clear and specific)
Get cloud functions operational
Implement a prototype which runs upon calling the function, with plans to deploy function to cloud which waits for the creation of a firestore document to activate
Write basis of LaTeX template
Call function to make a copy of the tex template to fill in for the invoice
Download the copy of the template.
Query into firestore database collection.
Take said document data and assign keywords that match on the document with LaTeX keywords for template and assign it to them.
Take the LaTeX code and put onto the LaTeX copy of template.
Call function which returns a promise of a pdf generation of the .tex document.
Call function which returns a promise an email sending of the generated pdf as an attachment

End of sprint assessment
We managed to complete several of the tasks, some to a further extent than others. We will continue to improve upon these and set them as tasks for the next sprint.

Sprint retrospective comments
After hearing Sam's recommendations for other groups as well as our, we are going to change such things as files saved in temp file in memory rather than. While a lot of things are working, they need to be revised to make a better functioning system. We also need put the information from firestore into one document.

Course name: csc 131-02

Sprint# 2

Team name: ENIAC

Team members:
Ayush Thapa, Bilal Baloch, Anthony Torres, Dylan Khon, Evan Selby

Sprint start date: 10-03-23
Sprint end date: 10-17-23

Sprint tasks and goals (be clear and specific)
Work on req & design
Update code to query nontabular firestore database structure
Develop a better LaTeX document (a better .tex file)
Develop code to query tabular firestore database
Develop code to write queried tabular data in LaTeX code

End of sprint assessment
We were able to structure the input data into one firestore document as opposed to our previous two. A better .tex document was also developed to look and serve more as a proper invoice. We were also able to create some functions to handle tasks using pub/sub and https requests more properly.

Sprint retrospective comments
On Sam's advice we will be straying away from pub/sub and trying to get our functions up and running with a different solution. We will also work on further updating aspects of the code to ensure functionality.

Course name: csc 131-02

Sprint# 3

Team name: ENIAC

Team members:
Ayush Thapa, Bilal Baloch, Anthony Torres, Dylan Khon, Evan Selby

Sprint start date: 10-17-23
Sprint end date: 10-31-23

Sprint tasks and goals (be clear and specific
Work on req & design
Get cloud functions operational without pub/sub
Add additional fields to the firestore and blank template
Continue working on improving the .tex document template
Get the functions to update the template correctly

End of sprint assessment
We managed to get the cloud functions operational without pub/subs and sending information to the template. The .tex document is updated with information and it also now able to handle text wrapping as well as continuing on to the next page if it has to.

Sprint retrospective comments
Things to be added, being able to see execution line by line of a program triggered by a firestore document created and processes the data from the trigger initially, how it saves names based on the documents invoice number. Error handling of cases where the document doesn't contain all the information, return back an error saying certain fields are not inserted. We will also continue making improvements to the .tex document including better wrapping and layout.

Course name: csc 131-02

Sprint# 4

Team name: ENIAC

Team members:
Ayush Thapa, Bilal Baloch, Anthony Torres, Dylan Khon, Evan Selby

Sprint start date: 10-31-23
Sprint end date: 11-14-23

Sprint tasks and goals (be clear and specific)
Main priority: Get latex to pdf conversion working with cloud function
possible solutions cloud Kubernetes with docker, pdf-lib, using other latex distributions
Continue updating Req & des as well as the code

End of sprint assessment
We tried several things to the and get the conversion to work but they seemed to not get any results. Were able to create a docker image with a tex distribution and then push it to cloud Kubernetes, this seemed promising, but we did not get it operational with the cloud function.

Sprint retrospective comments
Going off Sam’s advice we decided to not try and use cloud Kubernetes, but instead look into solutions that other teams were looking into like using an api, creating a npm file, or try and use a docker image.

Course name csc 131-02

Sprint# 5

Team name: ENIAC

Team members:
Ayush Thapa, Bilal Baloch, Anthony Torres, Dylan Khon, Evan Selby

Sprint start date: 11-14-23

Sprint end date:
11-28-23
Sprint tasks and goals(be clear and specific)
Continue trying to get a solution for the generation of the pdf either with docker and cloud run, npm, or api

End of sprint assessment
We tried getting each method to work, we tried npm at first but then switched mainly to a docker image containing a latex distribution that would run on cloud run. We have made progress in developing a functional image, but it does not yet work in conjunction with the rest of our code.

Sprint retrospective comments
After listening to other groups struggling with the same issue, we decided to try and implement some of their suggesting for the docker image and stray away from their mistakes. At the same time will also try and see if we could get it to work with dynamic docs api instead.