wolframBeta

Alan Chen, Julius Freyra, Stephanie Yoon, Angela Yu pd. 8
Final project

Overview:

Our goal is to make a tool that would be both useful and intuitive. Our idea is to make a linear algebra calculator that will give the user step-by-step solutions to otherwise time-consuming and tedious calculations. What makes it particularly intuitive to use is that the calculator will take input in three easy ways: LaTeX source code, uploaded pictures, and canvas drawings. Moreover, the calculator will not only display the steps in the solution, but it will also provide the LaTeX source code for each.

Tools:

- Backend Image Parser
  - http://docs.mathpix.com/#introduction
  - used to convert uploaded photo to LaTeX (takes an image file and returns a JSON, with key "latex" and value of image content in LaTeX format)
- Front End LaTex to HTML Converter
  - https://www.mathjax.org/
  - takes string in LaTeX format and renders in web page (either in HTML or SVG)
  - Mathjax.org-hosted CDN has been taken down but mathjax has presented alternative CDNs
    - https://www.mathjax.org/cdn-shutting-down/
    - CDNs:
      - https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?...
      - https://cdn.rawgit.com/mathjax/MathJax/2.7.1/MathJax.js

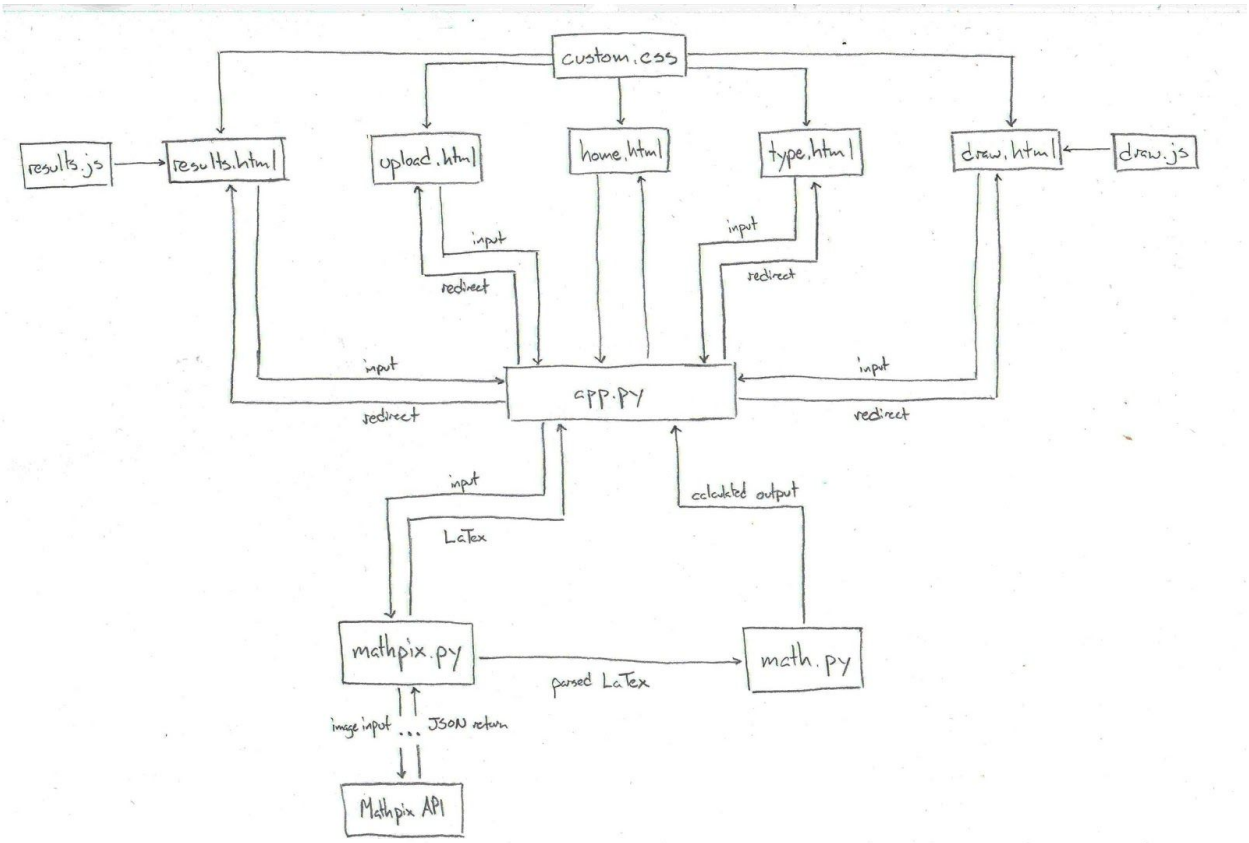
Component Breakdown:

- app.py
  - Flask App
  - Connects HTML

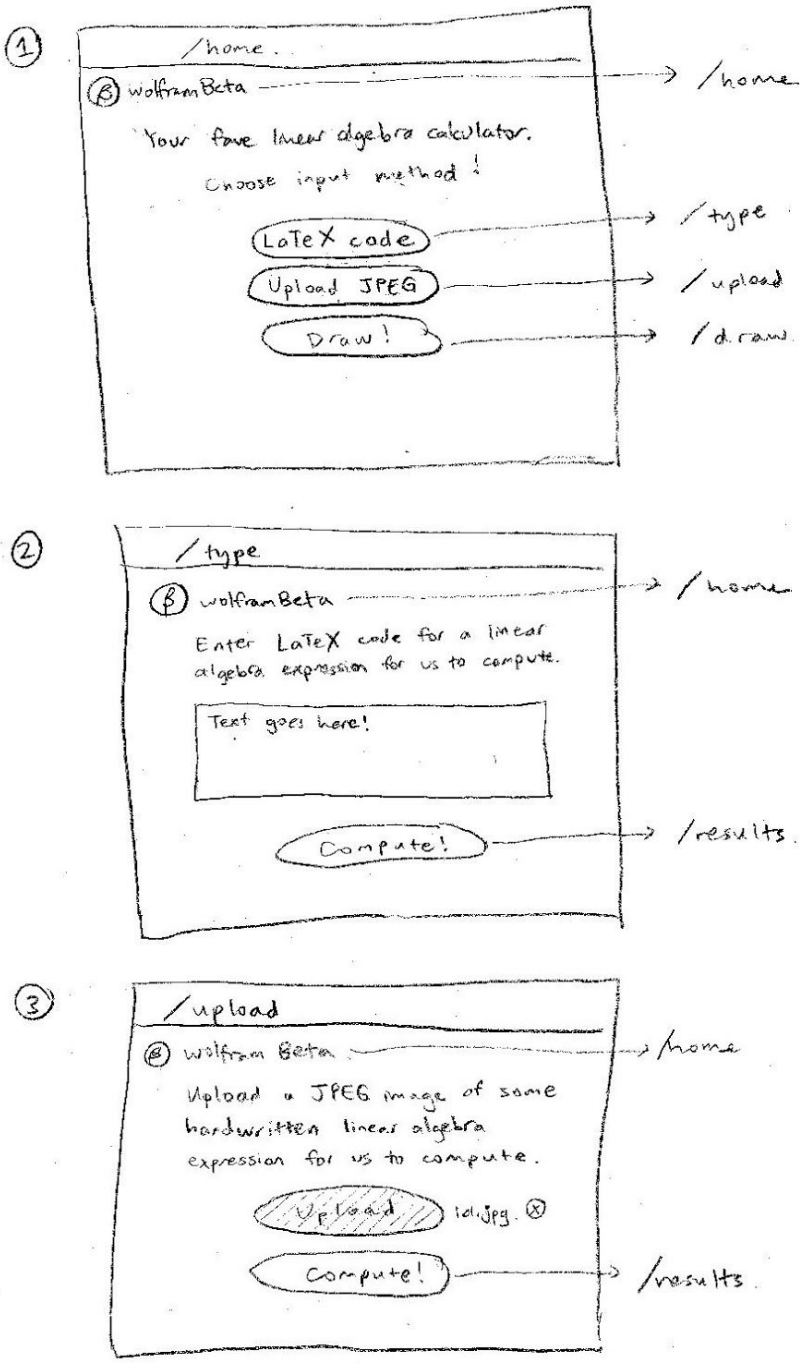
    - Facilitates image transfer between frontend and backend
- utils/
    - mathpix.py
        - Deals with Mathpix API
        - Sends images to API, parses returned JSON/LaTeX
    - math.py
        - Takes parsed LaTeX from JSON and calculates value or expression, then returns LaTeX back to Flask App (does this recursively to show the step-by-step solution)
        - function to convert LaTeX into HTML using mathjax
        - can use functional programming!
- static/
    - js/
        - draw.js
            - script for draw.html
            - canvas so that the user can use their mouse to draw
        - results.js
            - script for steps displaying LaTeX source codes
    - css/
        - custom.css
            - Makes things pretty
            - Using Bootstrap framework
            - Note: Creative Tim has some really pretty stuff! Should totally consider using it.
- templates/
    - home.html
    - upload.html
        - Single page for uploading images
    - draw.html
        - Page for drawing on canvas
        - submit => turns canvas drawing into a jpeg to run through mathpix
    - type.html
        - Page for typing LaTeX as input
    - results.html

- Displays step-by-step solution (card format: each step has a card, has option to display LaTeX source code)

Component Map:



Site Map:

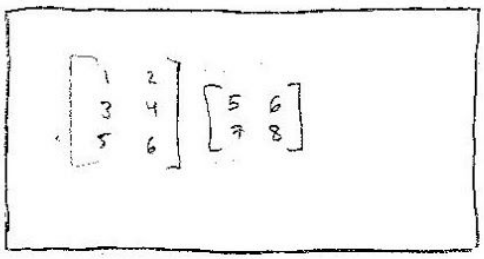

1
/home
WolframBeta
/home
Your fave linear algebra calculator.
Choose input method!
LaTeX code
/type
Upload JPEG
/upload
Draw!
/draw
2
/type
WolframBeta
/home
Enter LaTeX code for a linear algebra expression for us to compute.
Text goes here!
Compute!
/results
3
/upload
Wolfram Beta
/home
Upload a JPEG image of same handwritten linear algebra expression for us to compute.
Upload
id.jpg.
Compute!
/results

④

/draw

(E) wolfram Beta. → /home.

Click and drag to draw on this canvas a linear algebra expression for us to compute.



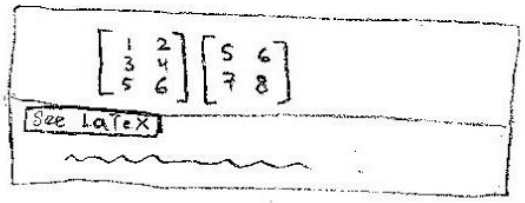
Compute! → /results.

⑤

/results

(3) wolfram Beta → /home.

You gave us:



Step-by-step solution:

See LaTeX

See LaTeX

The answer:

See LaTeX

Go back to home → /home.

Task Delegation:

- Steph (Project Manager)
  - math.py
- Alan
  - *.html
  - custom.css
  - math.py
- Julius
  - mathpix.py
  - app.py
- Angela
  - results.js
  - draw.js

Timeline:
(By) May 12:
- Julius + Angela: Design Doc

May 14:
- Alan: *.html, setup Bootstrap
- Angela: start on draw.js
- Julius: basic app.py, start on mathpix.py
  - redirections, calls
- Steph: basic operations ()

May 22:
- Alan: finished *.html, custom.css
- Angela: finished draw.js, start on results.js
- Julius: full structure app.py, finished mathpix.py
- Steph: more operations ()

May 28:
- Alan: any additional css, help with math.py
- Angela: finish results.js
- Julius: optimizing app.py & mathpix.py
- Steph: begin integration with mathpix.py
  - Parsing data, display

June 7: (soft deadline)
- All: integration, optimization, and debugging of all parts

June 10: (hard deadline)
- All: finishing touches