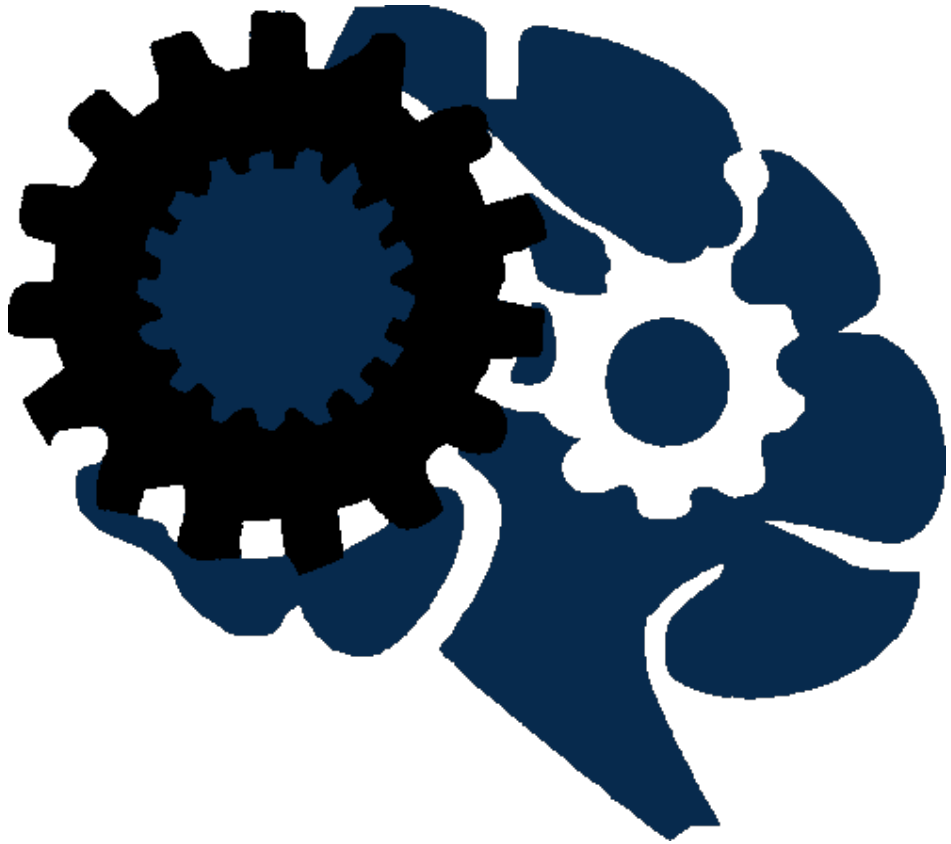


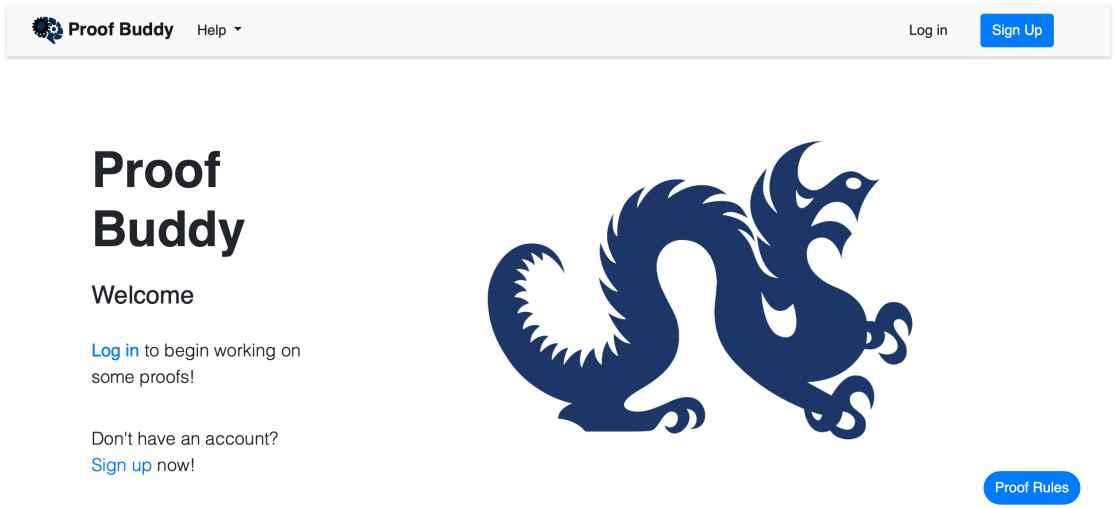
ProofBuddy

Meet the Team

Member	Role
Nicole Itchon	Team lead, QA lead
Raphael Perez	Front-end developer, UI/UX designer
Viet Pham	Back-end developer
Iftekhar Rahman	Back-end developer
Steve Earth	Stakeholder
Jeremy Johnson	Stakeholder

Description and Justification

- ProofBuddy is an educational tool for teaching computer science students proof techniques and logical reasoning
- Web-based and designed for use by both instructors and students
- Features include:
  - User authentication
  - Course creation and add students to course
  - Assignment and problem creation
  - Solve proofs and obtain feedback as you solve
- Currently does proofs in Boolean Logic and First Order Logic
  - Plans for extension to do Equational Reasoning



Currently available at http://bvm83.cci.drexel.edu

NICOLE

System Requirements

Functional requirements

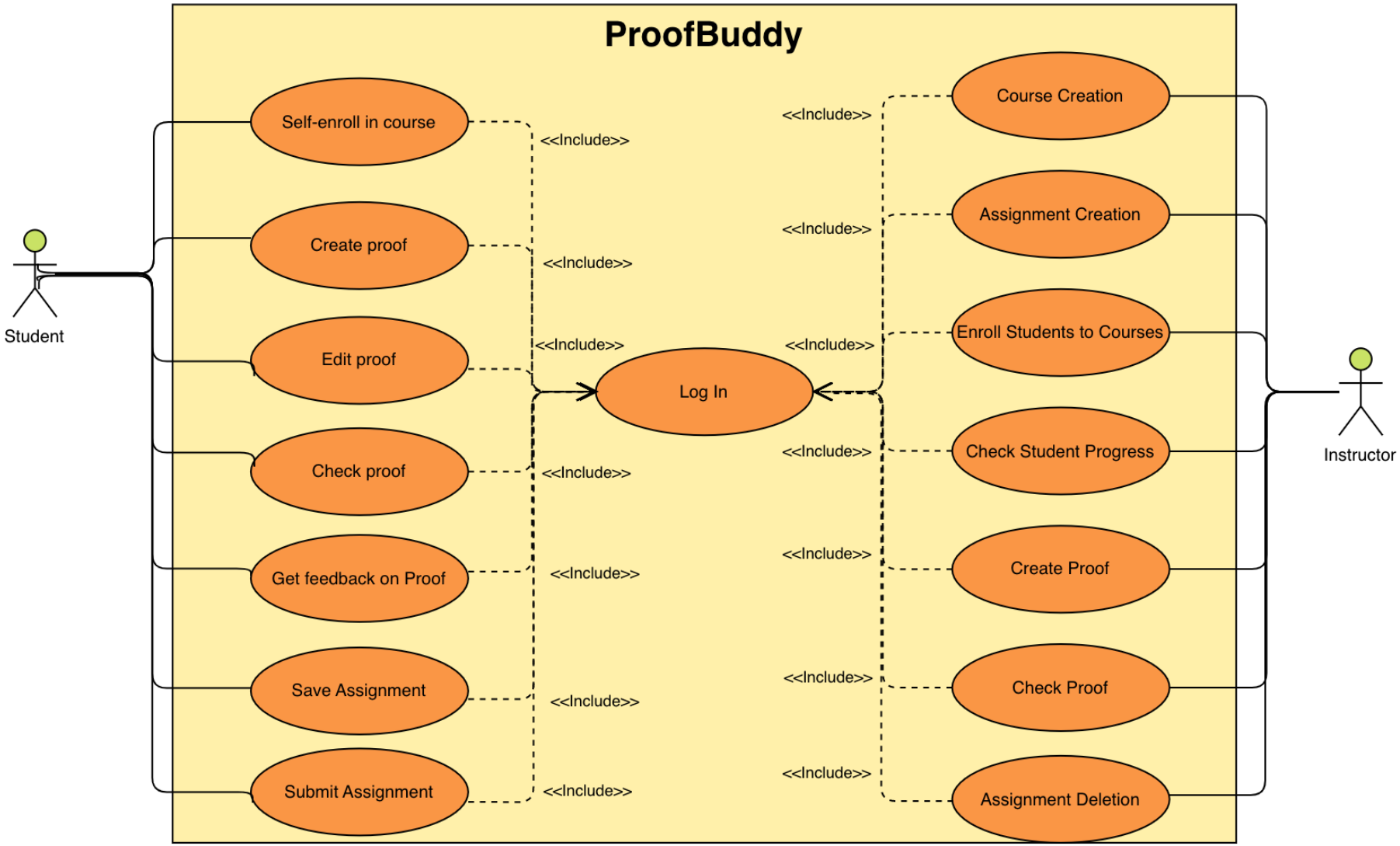
- Allow access to users with verified email address
- Store courses, assignments, and proofs
- Associate students with courses
- Associate problems and courses with assignments
- Verify that proofs are correct
- Provide feedback when proofs contain an error or is incomplete
- Grade assignments

Usability requirements

- Allow instructors to create/edit/delete courses, assignments, and problems
- Courses must have a title, term, section, and students
- Assignments must have a title, course, start date, due date, and problems
- Problems must have a title, points, target steps, lost points, rules, premise, and conclusion
- Allow users to restart, check, or save proofs
- Allow students to submit assignments
- Allow instructors to view student progress on proofs

NICOLE

Use Cases



IFTY

Architecture

4 Main Components (apps)

Accounts

- Account creation
- User Authentication
- Permissions

Courses

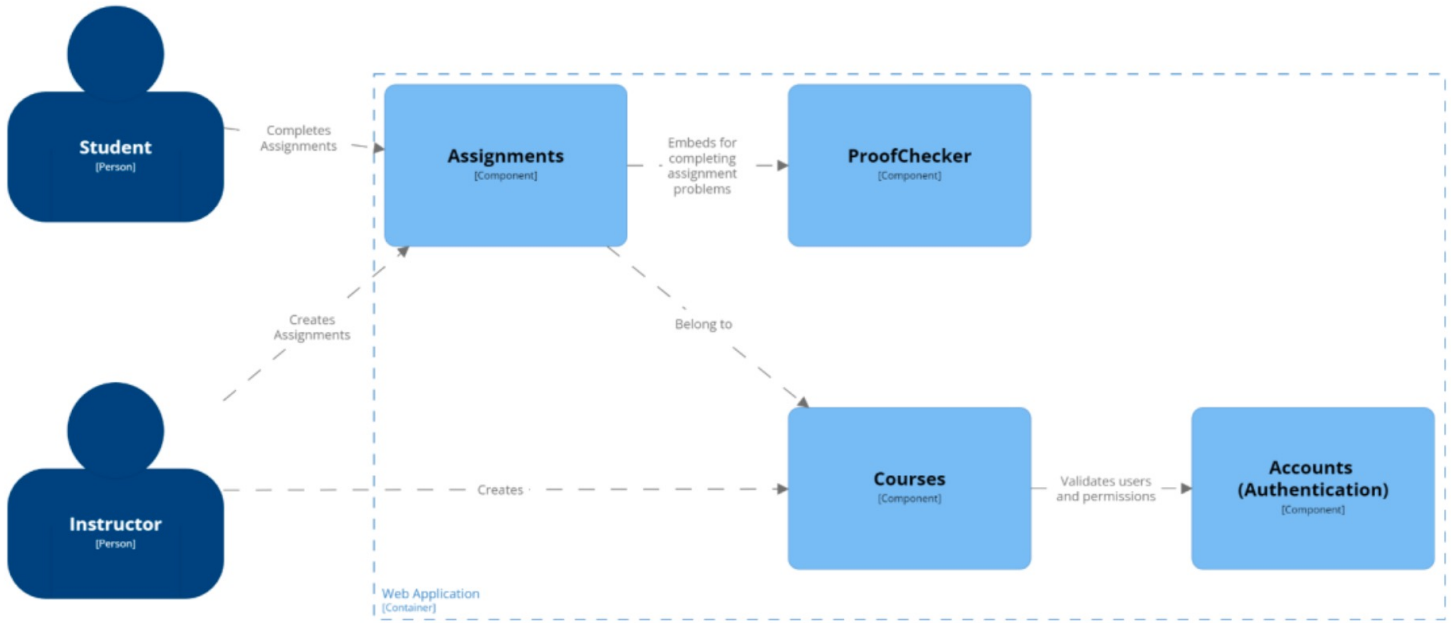
- Instructors create courses and enroll students
- Students self-enroll

Assignments

- Instructors create assignments for courses
- Students complete/submit assignments

ProofChecker

- Users create, edit, delete and save proofs
- Provides feedback and validation of the proofs submitted to the tool

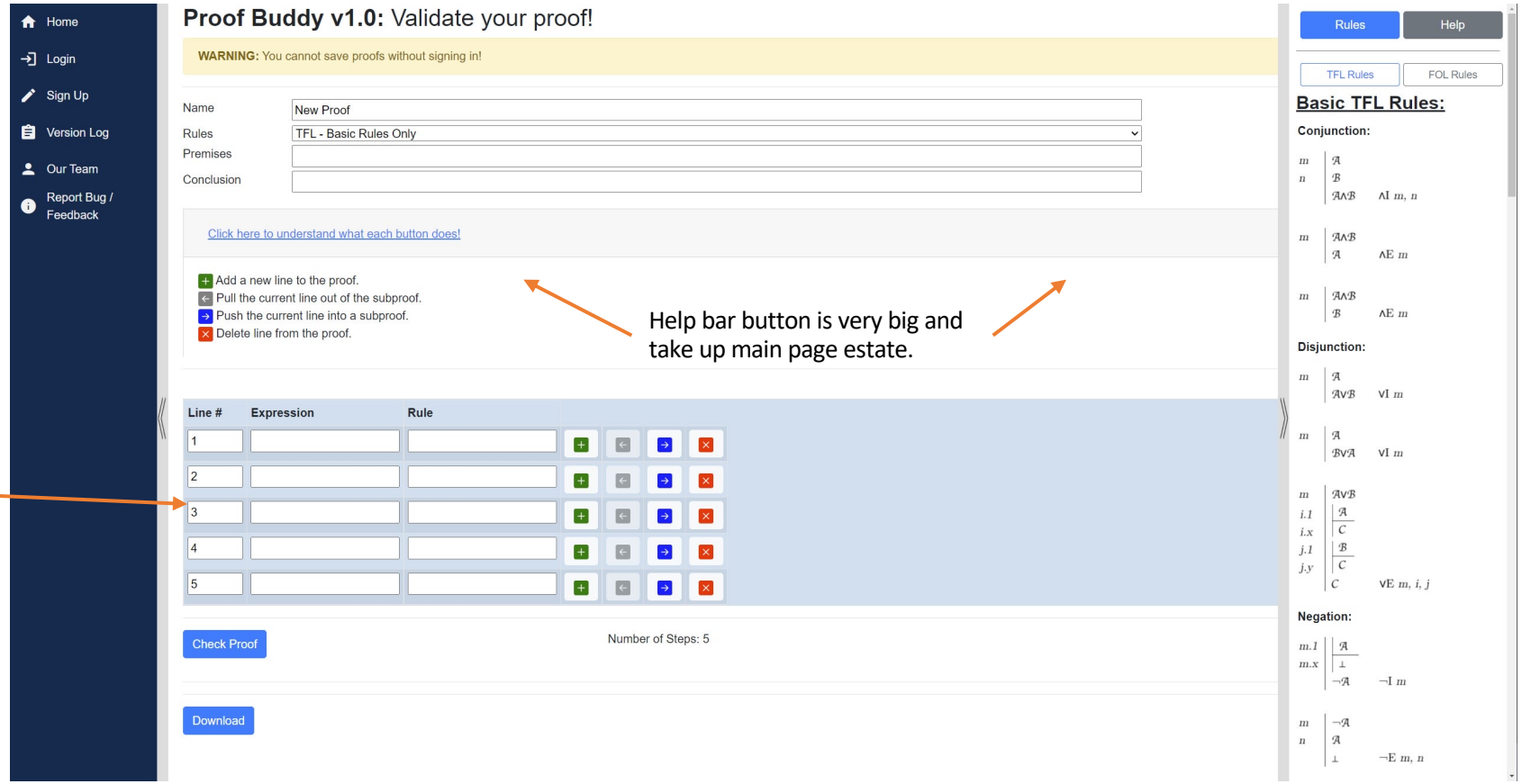


IFTY

Design and Implementation

The old design as shown was a UI for an app that mainly focused on the functionality of the app and added UI elements as a necessity over form.
The app is fully functional, but some UI elements can be improved.

Left-menu bar that is collapsed by clicking on the thin grey bar and takes up page estate.



Help bar button is very big and take up main page estate.

Proof buddy engine with some padding issues, the table will either run of page or have an unnecessary scroll bar and the line number are not editable though it appears to be in this version.

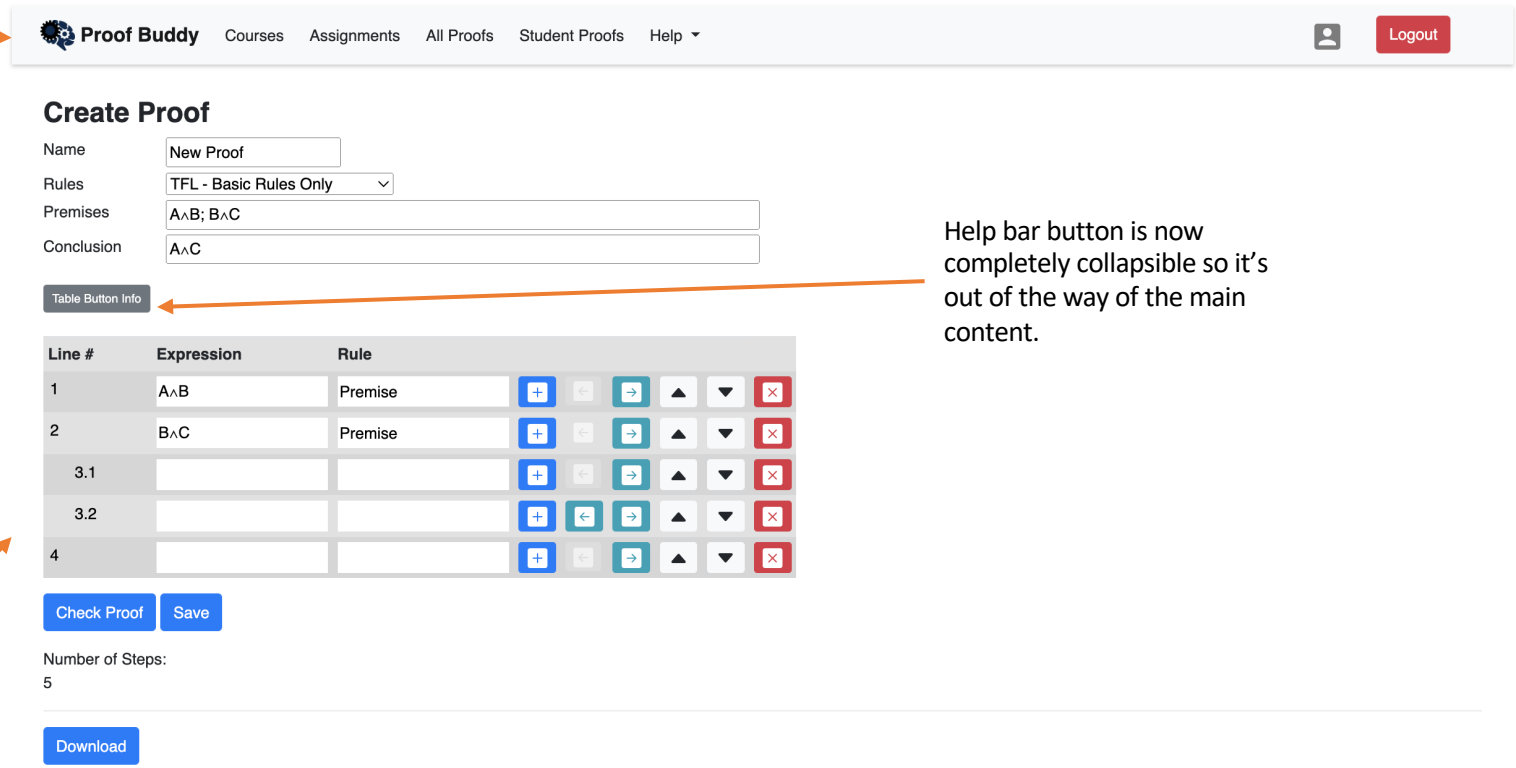
Right-menu bar that is *also* collapsed by clicking on the thin grey bar and takes up page estate. It's ok if this menu takes up space because there are a lot of rules and they're necessary for the assignments.

RAPH

Design and Implementation

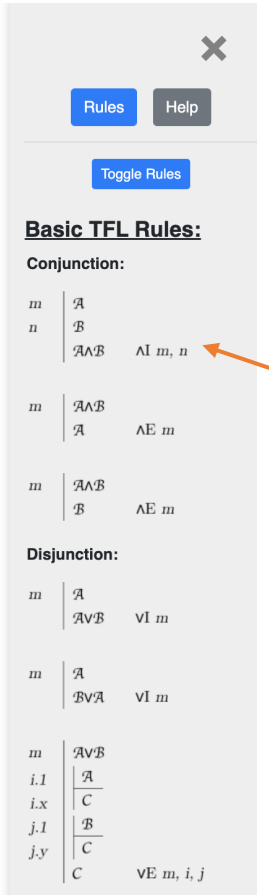
The new design is now more focused not only bringing the application to modern design standards, but also allowing the app to make more use of the web page estate to make the user find UI elements more easily.

Left-menu bar has been retooled as a sticky top nav bar that does not get in the way of any tables or text.



Help bar button is now completely collapsible so it's out of the way of the main content.

Proof buddy engine with padding fixes, clearer color definitions for the buttons and its functionality and line numbers do not appear editable for better UI clarity.

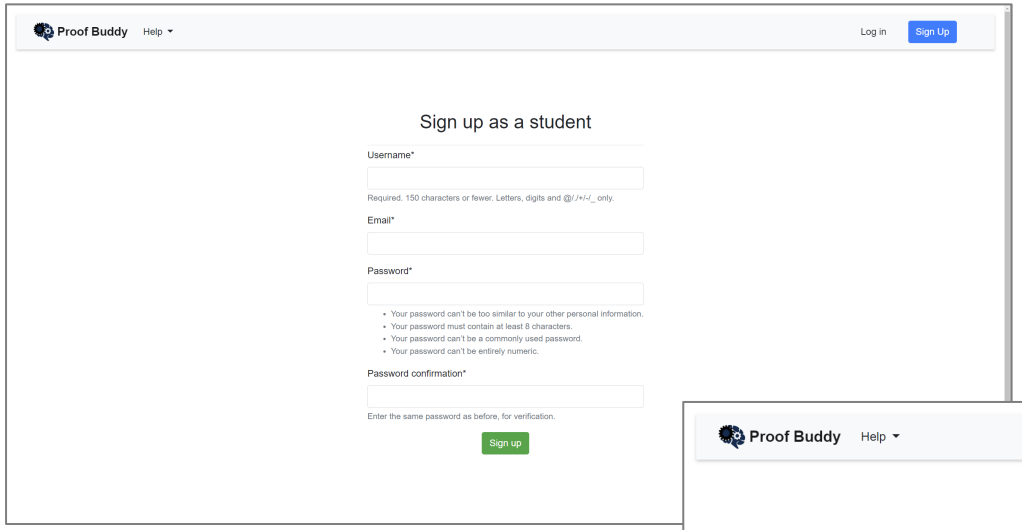


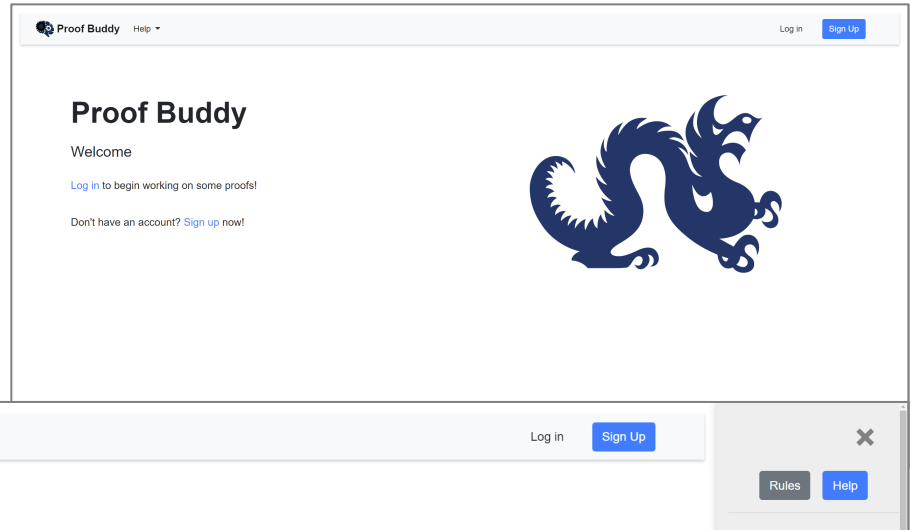
Right-menu bar now opened and closed with big, easy to find UI elements including the 'X' on top and users can now toggle rules on and off to only display the ones needed.

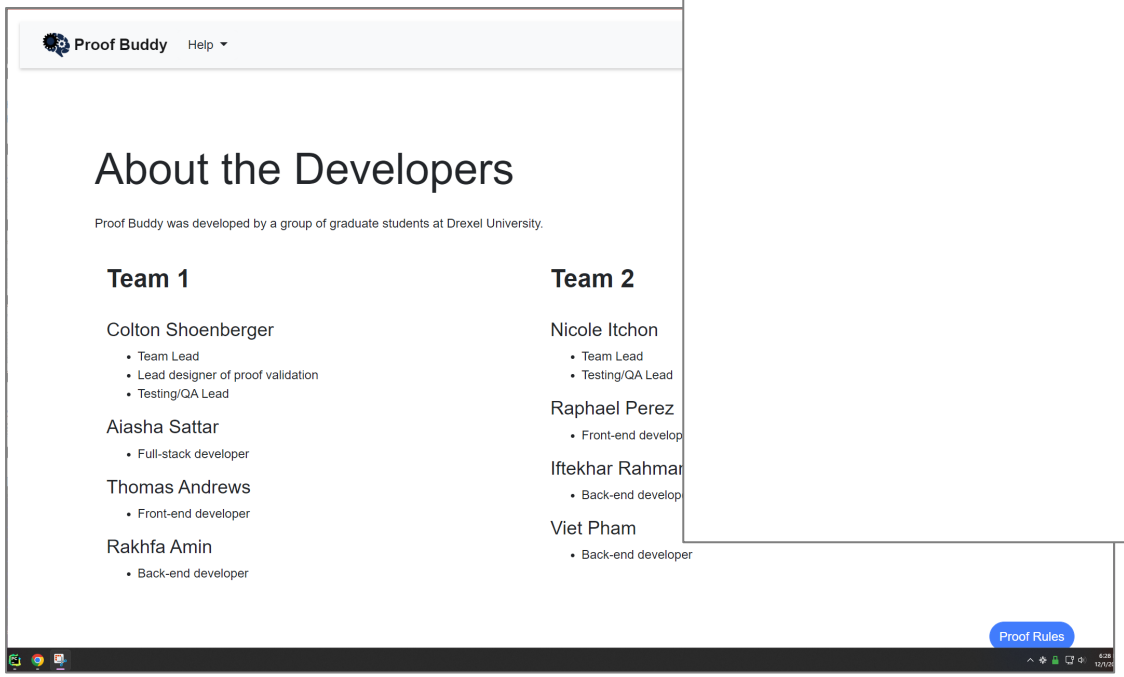
RAPH

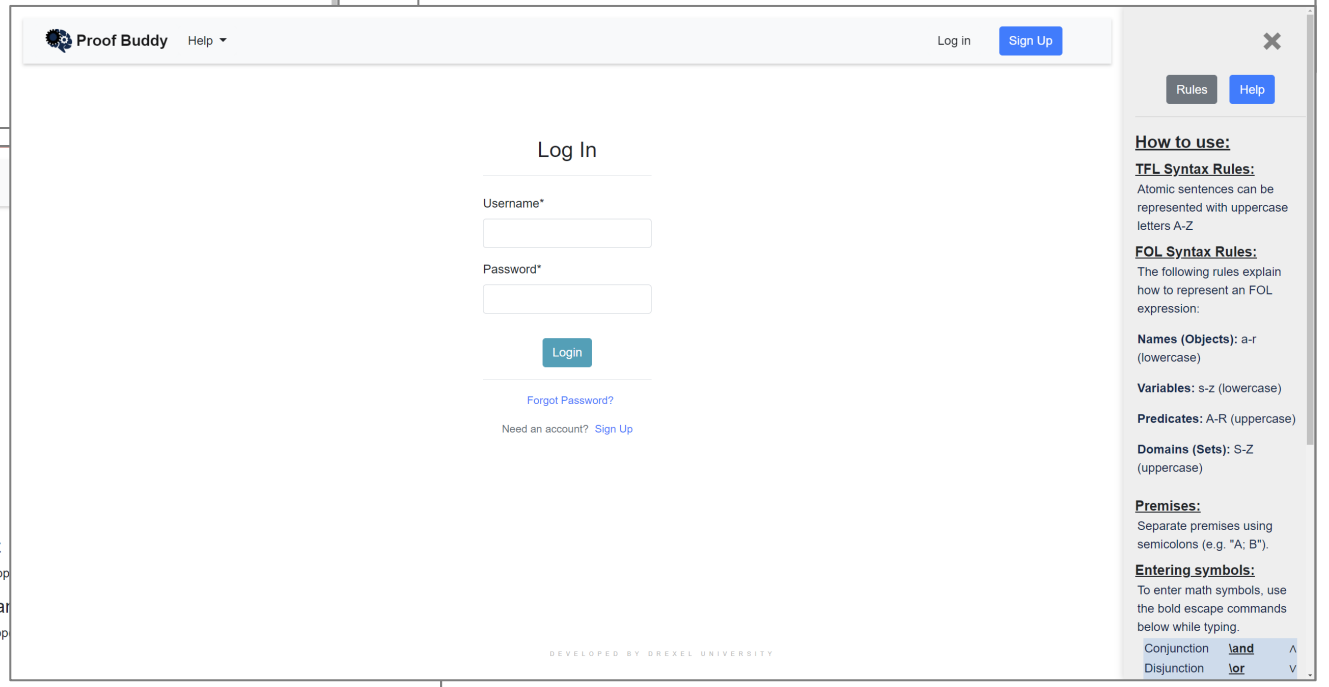
Design and Implementation

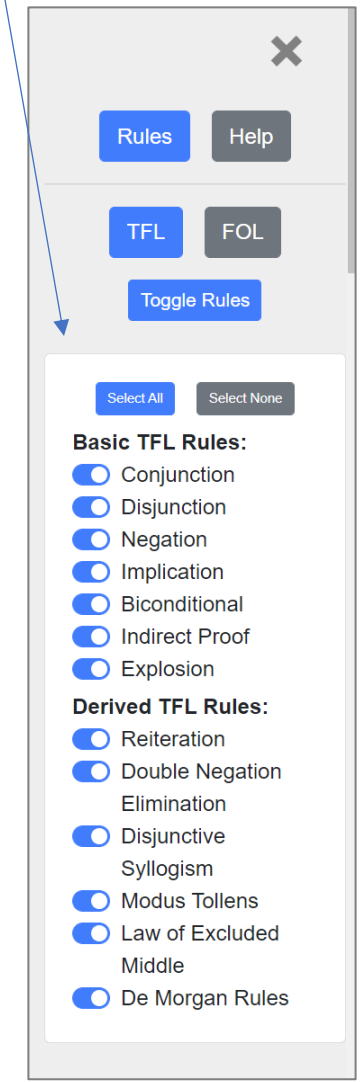
The part of the right-menu bar that allows users to hide rules they don't currently need to reduce clutter.











RAPH

Test Plan

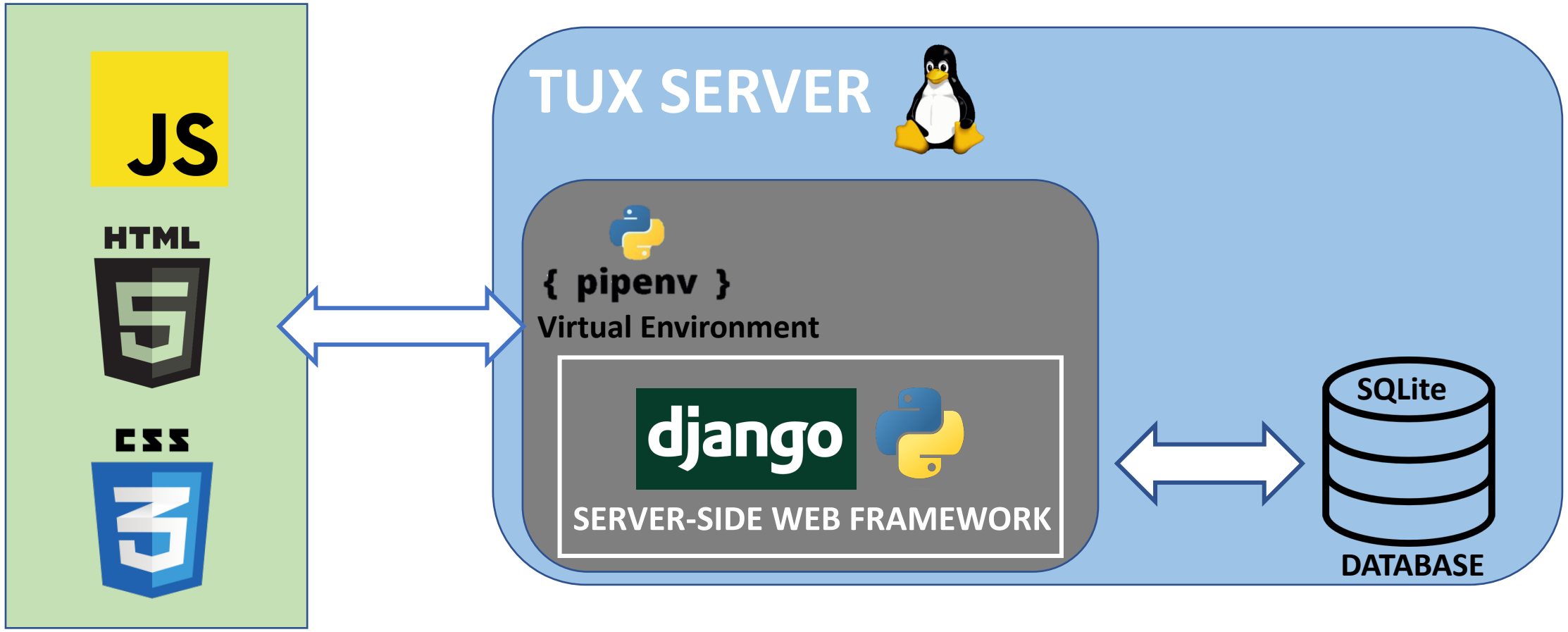
Currently

- Using **`unittest`** module, which is built-in to the Python standard library
- Can run **`python manage.py test`** within project to run existing tests
- Only tests basic functionalities and proofs

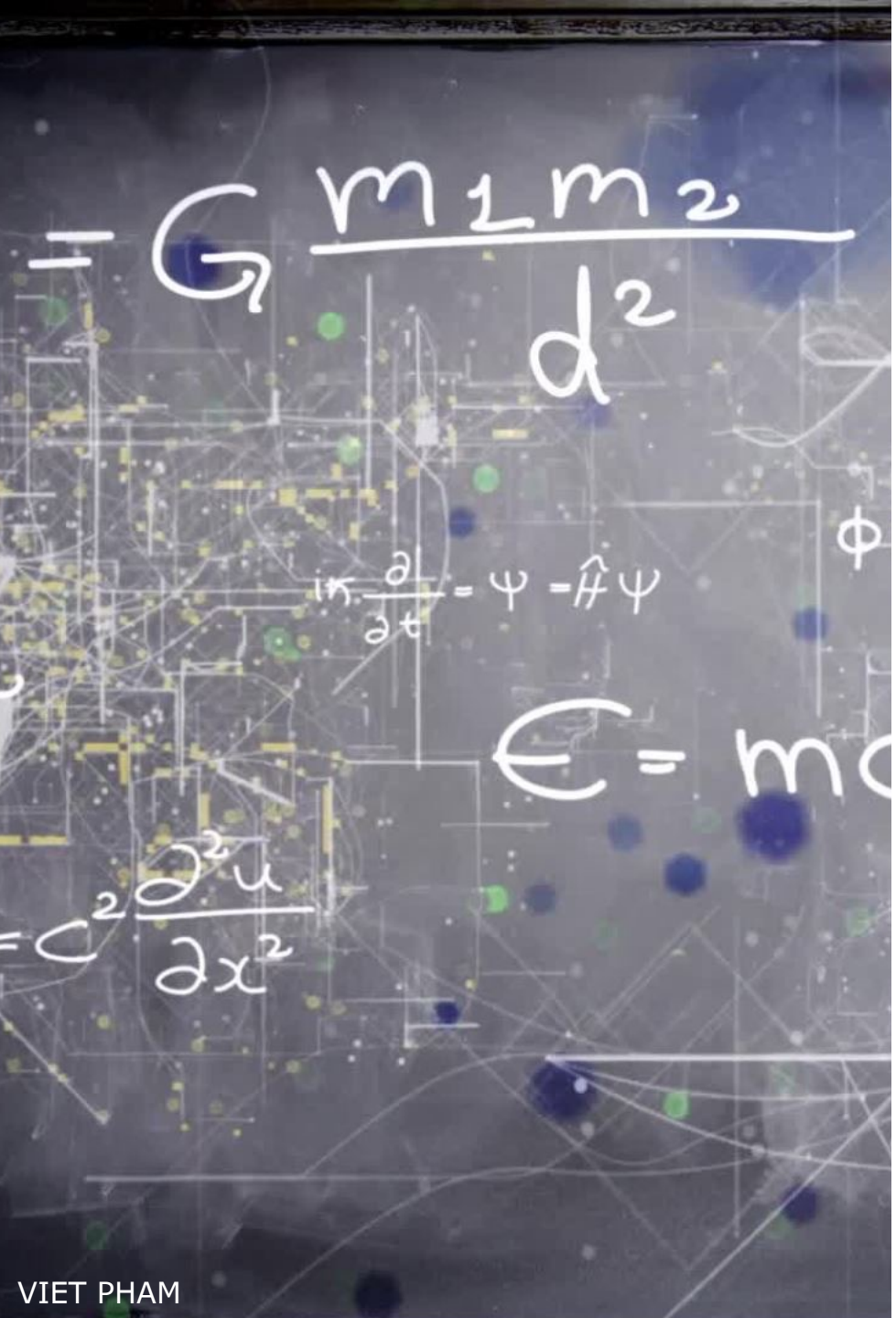
Improvements

- Extend testing as new functionalities are added
- Write tests to catch bugs at edge cases
- Implement continuous integration:
  - When developers create pull requests, have an automated build process that verifies code, runs test suite, and runs quality control checks

Supporting Technology - Framework



Supporting Technology - Others



Timeline and Projection for Next Term

- Framework for Equational Reasoning
- "Disprove" mode
- Proof by Induction
- And a few more...

ProofBuddy Demo