

ProofBuddy

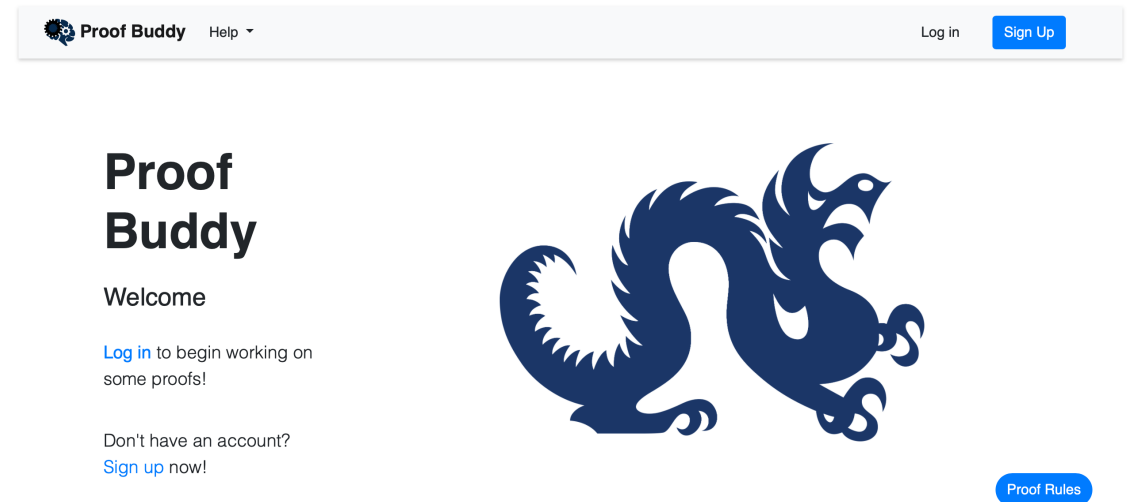


## Meet the Team

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>

# System Requirements

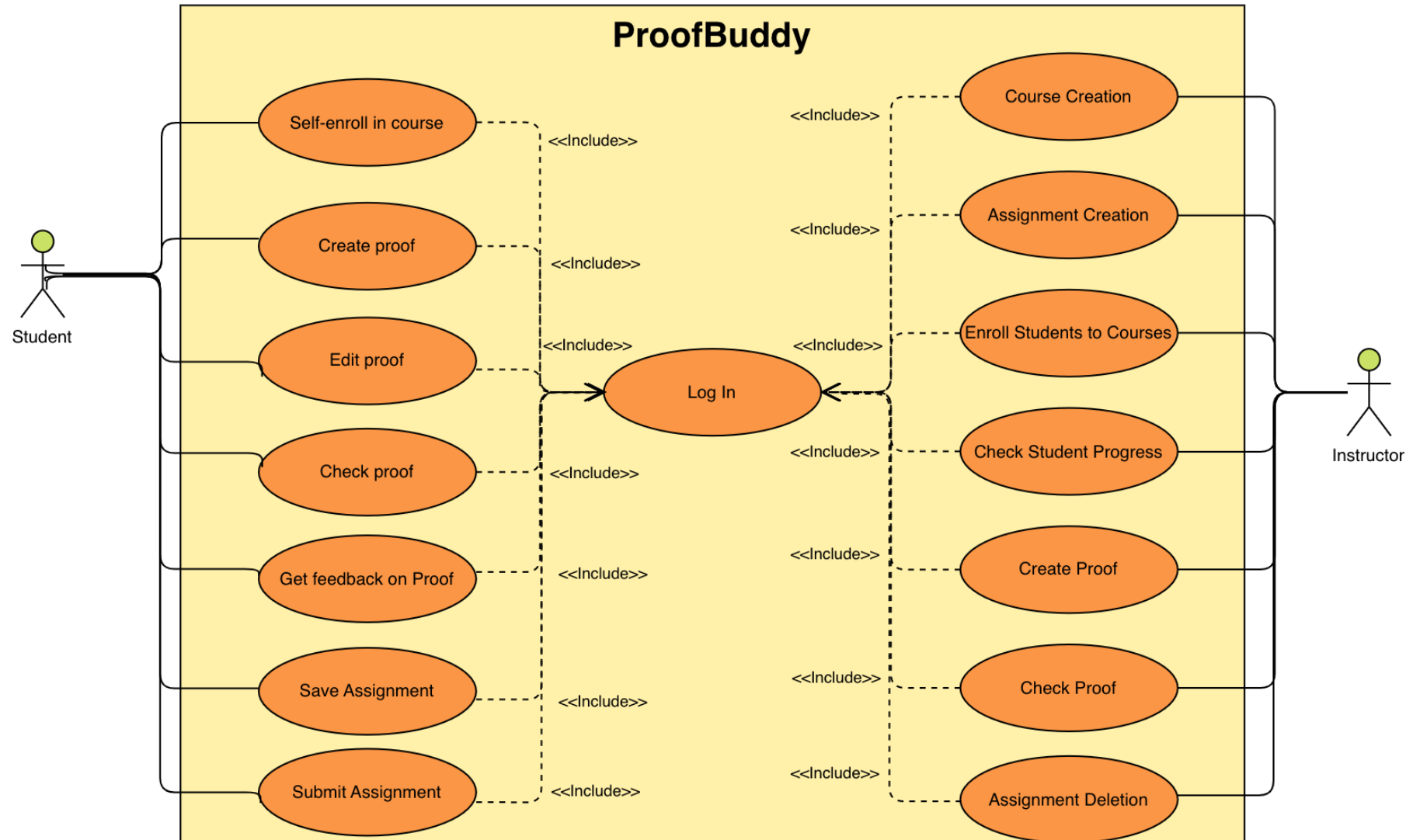
## Functional requirements

- 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

# Use Cases



# Architecture

## 4 Main Components (apps)

### Accounts

- Account creation
- User Authentication
- Permissions

### Courses

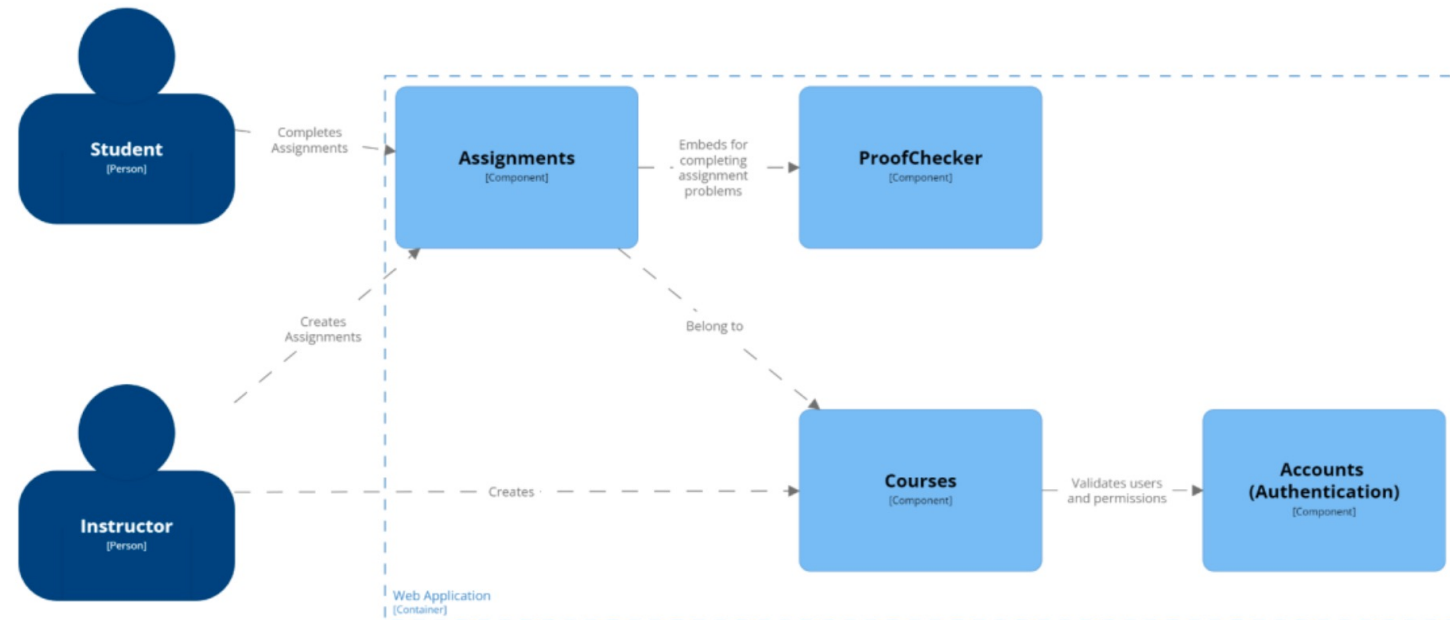
- Instructors create courses and enroll students
- Students self-enroll

### 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



# 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.

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.

**Proof Buddy v1.0: Validate your proof!**

**WARNING:** You cannot save proofs without signing in!

Name:

Rules:

Premises:

Conclusion:

[Click here to understand what each button does!](#)

**Help:**

- + Add a new line to the proof.
- Pull the current line out of the subproof.
- Push the current line into a subproof.
- × Delete line from the proof.

Line #	Expression	Rule
1	<input type="text"/>	<input type="text"/>
2	<input type="text"/>	<input type="text"/>
3	<input type="text"/>	<input type="text"/>
4	<input type="text"/>	<input type="text"/>
5	<input type="text"/>	<input type="text"/>

Number of Steps: 5

**Rules** **Help**

**Basic TFL Rules:**

**Conjunction:**

- $m \quad A$   
 $n \quad B$   
—————  
 $A \wedge B \quad \wedge I \ m, n$
- $m \quad A \wedge B$   
—————  
 $A \quad \wedge E \ m$
- $m \quad A \wedge B$   
 $n \quad B$   
—————  
 $A \quad \wedge E \ m$

**Disjunction:**

- $m \quad A$   
—————  
 $A \vee B \quad \vee I \ m$
- $m \quad A$   
 $n \quad B \vee A$   
—————  
 $A \quad \vee I \ m$
- $m \quad A \vee B$   
 $i.I \quad A$   
 $i.x \quad C$   
 $j.I \quad B$   
 $j.y \quad C$   
—————  
 $C \quad \vee E \ m, i, j$

**Negation:**

- $m.I \quad A$   
 $m.x \quad \bot$   
—————  
 $\neg A \quad \neg I \ m$
- $m \quad \neg A$   
 $n \quad A$   
—————  
 $\bot \quad \neg E \ m, n$


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.

# 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.

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.

 **Proof Buddy** Courses Assignments All Proofs Student Proofs Help ▾

Logout

### Create Proof

Name

New Proof

Rules

TFL - Basic Rules Only ▾

Premises

A ∧ B; B ∧ C

Conclusion

A ∧ C

Table Button Info

Line #	Expression	Rule						
1	A ∧ B	Premise	+	←	→	▲	▼	×
2	B ∧ C	Premise	+	←	→	▲	▼	×
3.1			+	←	→	▲	▼	×
3.2			+	←	→	▲	▼	×
4			+	←	→	▲	▼	×

Check Proof

Save

Number of Steps:

5

Download

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

×

Rules Help

Toggle Rules

#### Basic TFL Rules:

**Conjunction:**

m		A		
n		B		
		A ∧ B	∧ I m, n	
m		A ∧ B		
		A	∧ E m	
m		A ∧ B		
		B	∧ E m	

**Disjunction:**

m		A		
		A ∨ B	∨ I m	
m		A		
		B ∨ A	∨ I m	
m		A ∨ B		
i.1		A		
i.x		C		
j.1		B		
j.y		C		
		C	∨ E m, i, j	

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.



# Design and Implementation

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

Proof Buddy

Help

Log in

Sign Up

Sign up as a student

Username\*

Required: 150 characters or fewer. Letters, digits and @/./+/\_ only.

Email\*

Password\*

Your password can't be too similar to your other personal information.

Your password must contain at least 8 characters.

Your password can't be a commonly used password.

Your password can't be entirely numeric.

Password confirmation\*

Enter the same password as before, for verification.

Sign up

Proof Buddy

Help

Log in


Sign Up

Proof Buddy

Welcome

Log in to begin working on some proofs!

Don't have an account? Sign up now!



Proof Buddy

Help

Log in

Sign Up

Log In

Username\*

Password\*

Login

Forgot Password?

Need an account? Sign Up

DEVELOPED BY DREXEL UNIVERSITY

Rules

Help

How to use:

TFL Syntax Rules:

Atomic sentences can be represented with uppercase letters A-Z

FOL Syntax Rules:

The following rules explain how to represent an FOL expression:

Names (Objects):

a-r (lowercase)

Variables:

s-z (lowercase)

Predicates:

A-R (uppercase)

Domains (Sets):

S-Z (uppercase)

Premises:

Separate premises using semicolons (e.g. "A; B").

Entering symbols:

To enter math symbols, use the bold escape commands below while typing.

Conjunction

**land**

^

Disjunction

**lor**

v

Rules

Help

TFL

FOL

Toggle Rules

Select All

Select None

Basic TFL Rules:

Conjunction

Disjunction

Negation

Implication

Biconditional

Indirect Proof

Explosion

Derived TFL Rules:

Reiteration

Double Negation

Elimination

Disjunctive Syllogism

Modus Tollens

Law of Excluded Middle

De Morgan Rules

About the Developers

Proof Buddy was developed by a group of graduate students at Drexel University.

Team 1

Colton Shoenberger

- Team Lead
- Lead designer of proof validation
- Testing/QA Lead

Aiasha Sattar

- Full-stack developer

Thomas Andrews

- Front-end developer

Rakhfa Amin

- Back-end developer

Team 2

Nicole Itchon

- Team Lead
- Testing/QA Lead

Raphael Perez

- Front-end developer

Iftexhar Rahman

- Back-end developer

Viet Pham

- Back-end developer

Proof Rules

# Test Plan

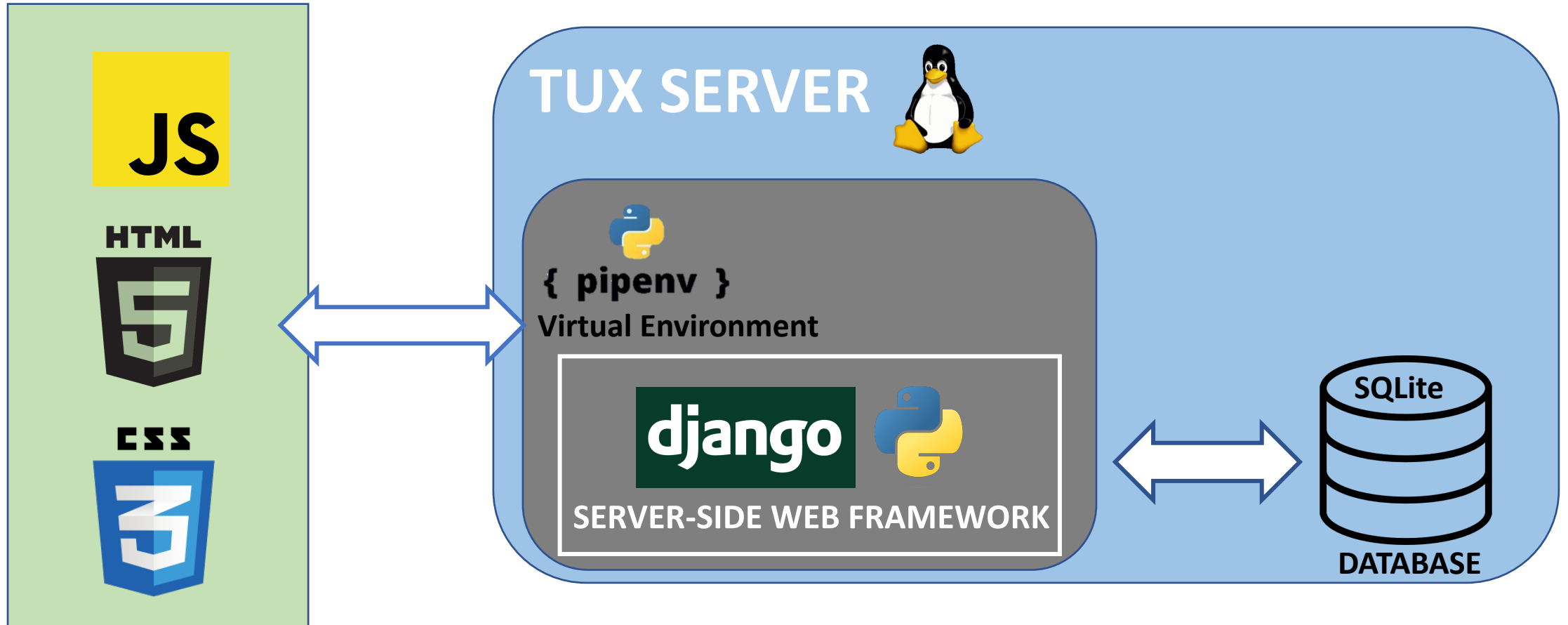
## Currently

- 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