Purpose and Goals:
Students learn best by learning from their peers and teaching each other. Current options let students learn from experienced teachers, but nowhere can students learn from other students and then help their peers. We want to provide a platform where:
- students of any age and discipline can learn from others at their same level of knowledge
- learners can pass on their knowledge and teach others, in the process gaining confidence in their own skills
- student-teachers can get feedback from their own students and improve upon their teaching

Context Diagram:
Key Concepts:

- **Lesson.** Teaches a particular subject using video, text explanation, or examples.
- **Subject.** Topic of knowledge under which lessons are categorized.
- **Review.** Rating system for the quality of lessons.
- **Video:** A portion of a lesson hosted on youtube.
- **Document:** A portion of a lesson consisting of text, hosted on Google Drive.

Data Model:

![Data Model Diagram](image)

Additional Notes on Data Model:

- Users cannot make reviews for their own lessons
- We store whether or not a lesson has been viewed by each user

Features:

- **View lessons** by searching for them by topic.
- **Create lessons** with videos and documents to prove mastery of the subject.
- **Review** the lessons you take with a 5-star system.
- **Get feedback** for the lessons you create via additional text notes on reviews.
Security concerns:

- Fake accounts can be created and used to spam advertisements.
  - Use Google authentication with Google accounts, ensuring users cannot make fake accounts on our website but must go through Google, which is far more secure.
- Links to “lessons” can actually point to other web pages or videos of inappropriate content.
  - Limit links to youtube and google drive to at least avoid links to pornography or viruses. Using Google auth lets us assure that links point to content created by the user.
- Users could try to embed javascript or SQL queries within the description text or link of a lesson for SQL injection, XSS, or CSRF attacks.
  - Never evaluate or do anything with lesson text other than displaying it, without parsing it as HTML, to prevent XSS attacks.
  - Use the safe querying tools provided by rails, which prevent SQL injection.
  - Use the `protect_from_forgery` rails attribute to prevent CSRF attacks by authenticating all POST forms with special tokens.
- Spamming reviews, positive or negative, to skew the rating system in the attacker’s favor
  - Set a limit on how many reviews can be made per user per time period.

Threat Model:

- People wanting to spam advertisements to our users
  - Mitigated by limiting the types of links allowed
- People wanting to view lessons without making any to take advantage of the site
  - Make users create a lesson per every x lessons they view for some x. Using Google auth ensures users can’t make new dummy accounts to keep viewing lessons
- People wanting to post inappropriate content to minors using the site
  - Again, limiting the types of links mitigates this. Also using a google account adds accountability if a user posts something inappropriate or illegal.
- Generic attackers looking for usernames and passwords
  - Don’t actually store usernames or passwords, just trust Google auth and stores unique (but essentially worthless) UIDs.
Wireframe Diagrams:

Log In:

TEACH IT FORWARD

Students learn best by learning from their peers and teaching each other. We want to provide a platform where students of any age and discipline can come together to learn from others at their same level of knowledge, then learn mastery of the material by passing on the knowledge to others.

Log in using your Google account

Don't have a Google account? Sign up now!

Lesson List:

View by Subjects

- Subject 1
- Subject 2
- Subject 3
- Subject 4

Created by: User X
Lesson X: Learn how to X

Created by: User Y
Lesson Z: Learn how to Z

Created by: User X
Lesson Y: Learn how to Y

Created by: User Z
Lesson Q: Learn how to Q
Lesson View Page:

Lesson: Software Development
Created By: Daniel Jackson

What did you think of this lesson?

★ ★ ★ ★ ★

Back to subject listing  Make your own lesson

Lesson Creation Page:

Teach it Forward

Creating a new Lesson

Enter title

Select from existing subjects

...or make your own subject

YouTube video url

Google Drive document url

Additional Text

Create

Back
Home/User Profile:

Your Lessons

Lessons You've seen

Lessons you've created:

Lesson W
View Reviews

Lesson X
View Reviews

Lesson Y
View Reviews

Lesson Z
View Reviews

Logout

View My Profile
Design Challenges:

- Storing the lessons (video, documents)
  - Options:
    - Store everything on Amazon Web Services. Pro: versatile. Con: more complicated
    - Use Google APIs to store the materials in Youtube and Google Drive. Pro: easy to use, popular on the web. Con: limited to youtube videos and drive documents.
  - Solution:
    - Use Google APIs because of the popularity such systems have gained and their ease of use. Additionally, this will allow us to use Google authentication to assure user content is actually theirs.

- User authentication
  - Options:
    - Traditional sign up/log in developed for application. Pros: keep everything internal. Cons: not as secure.
    - Use Google authentication. Pros: secure, users can use their existing accounts. Cons: have to deal with Google API.
  - Solution:
    - Using Google authentication will help us to user generated content is actually theirs and not copied from some other source.

- Lesson subjects
  - Options:
    - Have a predefined list of subjects for which lessons can be created. Pros: stops users from making fake subjects. Cons: less expandability.
    - Allow users to create new lesson subjects. Pros: easily expandable system. Cons: users can make new subjects when unnecessary.
  - Solution:
    - Allow users to create their own subjects, perhaps with topic and subtopic, but like Piazza does with questions, ask users to check that there isn’t already an equivalent topic worded differently.

- Lesson creation
  - Options:
    - Only allow users to create lessons for a particular subject if they have already viewed lessons on that subject. Pros: lessons all come from experienced users. Cons: users with prior experience but who haven’t seen lessons can’t make their own right away, and can’t make lessons for subjects that don’t yet exist.
Solution:
- In order to boost the quality of lessons, we only want to allow users to create lessons for a particular subject if they have already viewed lessons on that subject, if lessons exist for that subject. If no lessons exist, allow users to make a new lesson. This allows our subject list to expand.

Lesson Rating System
- Option:
  - Use upvote/downvote system. Pros: Easy to use and implement. Cons: review either all bad or all good, no in between.
  - Use 5-star system. Pros: can specify degrees of good/bad. Cons: slower for users to make reviews.
- Solution:
  - In order to get a more unbiased rating from the user's we decided to omit other user's rating and allow the user to make his own decision on the lesson and be unaffected by the "hive mind".

MVP Data Model Subset:
***it is the same as for the complete thing. all components are in place***