

Computer Science and Engineering Department, University of
Nevada, Reno

Lecture Goggles

Team #22, Logan Long, Nathan Yocum, Zachary Johnson

Dr. Sergiu Dascalu, Devrin Lee

Dr. Shamik Sengupta

December 13, 2018

Table of Contents

Table of Contents	2
Abstract	3
Introduction	3
Prototype Objectives & Functionality	4
Develop Prototype	8
Frontend Prototype	8
Evaluate Prototype	16
Demo Prototype	16
Changes Needed to Software Design	17
Contributions of Team Members	18

Abstract

The project proposed by Team 22 is a free, open-source, educational resource repository to help students gain a better understanding of school subjects. Lecture Goggles is meant to help resource sharing between students and professors. Initial implementation will be focused on a web application design using a back-end web server and database to quickly manipulate and store uploaded resources and user accounts. Lecture Goggles limits the administrator's interaction using a front-end design mimicking a tree structure that can be built upon as more resources and subjects are added.

Introduction

Lecture Goggles is a web application that is built to store and share helpful resources. Resources can be HTTPS links or files provided by students or teachers. This will be limit file sizes to an appropriate size which has not been determined yet. To provide a safe environment for all users, each link will be checked for malicious content or inappropriate HTTPS forwards. The first implementation will limit resource contribution to authorized college students and college professor accounts. To authorize accounts, the application will use third-party SheerID software to get access to 180+ million student verified email addresses from the US and around the world.

The UI design behind Lecture Goggles is an expanding tree method. By using this method of design Lecture Goggles can continue to grow with little administration interaction. As users add new subjects and resources to the website, the website will grow to contain all academic subjects needed. Each resource link will afford a system to upvote and downvote. As demonstrated in websites like reddit.com, this seems highly successful at increasing visibility to useful additions.

Due to the size, scope, and dynamic capability of Lecture Goggles, there will need to be a web server to communicate with each users' requests. Django is a web framework that can be used as an intermediary between the user requests and database. Although Django can also be the web server for Lecture Goggles, for easier deployment and scalability Team 22 will be using Apache Web Servers to contain Django.

To contain and organize all the user account data, provided resources, and any other necessary data, a database will be needed. MongoDB, which is classified as a NoSQL

database program, was previously going to be used for initial implementation. Because Django does not support NoSQL out of the box, our group will need to use a PostgreSQL database instead. Django web framework will access the database directly when user requests are received then communicate pertinent information back to the user.

The project will run with multiple servers communicating to provide separation between the project. Firstly, there will be a web server to communicate with the https requests sent from individuals on the web. The web server will also send requests to the API server to provide dynamic webpage interaction with the user. The API server is the second server in the chain, communicating with both the web server and the database server. Lastly, the database server will communicate directly to and only to the API server. Each of these servers will be running with an Ubuntu operating system using Azure server hosting. Originally the servers were going to use Windows 10 as the operating system, this was changed to reduce the cost of Azure hosting. Using Ubuntu on Azure is approximately 7 cents an hour.

Prototype Objectives & Functionality

The items we decided to implement for the prototype were:

1. The cloud architecture of the web application.
2. The front end (web server)
3. The public facing API server code
4. The private database schema

The cloud architecture is hosted in Microsoft Azure. There are currently 3 Ubuntu servers, placed inside of a virtual network. Each of the servers resides in its own subnet, with strict firewall rules that restrict the allowed communication in and out of the subnet. We chose to prototype this because the cloud servers are necessary for the rest of the project to be deployed.

The front end was prototyped using Adobe XD. Some of the prototype was then converted into code, and deployed to the web server using Apache HTTP Server. The server accepts HTTP and HTTPS traffic, but will be limited to only HTTPS traffic when we purchase an SSL certificate.

The public facing API server code was developed locally on Zachary's machine, as we had some issues deploying it to Azure. The API code is written on top of the Django REST framework. It will be deployed on a public facing server that only accepts HTTPS

traffic. This framework will allow the web server to communicate with the database server, as well as allow outside developers to interface with the database.

The database schema was deployed to the database server using PostgreSQL according to the outline in Project 3. The server is configured to only accept incoming PostgreSQL traffic and does not have the ability to communicate with the outside internet. We decided to enter the tables for prototypes so that the API could be developed further over the winter break.

The tables below provide a detailed breakdown of the prototype objectives for Lecture Goggles.

Table 1: Key

x	Implemented
-	In prototype but not implemented
	Not implemented or prototyped

Table 2: Frontend Prototype Objectives

Goal Description	Prototyped?
Home page	x
Sign In	x
Sign Up	x
Subject View	-
Topic View	-
Resource View	x
Resource Card	x
Upvoting and Downvoting Posts	-
404 (NotFound) page	x
Support Page	-
Developer Page	x
Hosted on server	x

Routing on server	
-------------------	--

Table 3: Architecture Prototype Objectives

Goal Description	Prototyped?
Web Server	x
API Server	x
Database Server	x
Storage Server	
Virtual Network	x
Subnets	
Web	x
API	x
Database	x
Storage	
Firewalls	
Web	x
API	x
Database	x
Storage	

Table 4: Database Prototype Objectives

Goal Description	Prototyped?
Schema	x
Tables	x
Account	x
Submissions	x

Subscriptions	x
Votes	x
Reports	x
Communication with API Server	-

Table 5: API Prototype Objectives

Goal Description	Prototyped?
Django Install	x
Models	
User	x
JSON translation	x

Develop Prototype

Frontend Prototype

Figures 3, 4, 5, 6, 7, 8, and 9 display the frontend prototype designed with Adobe XD. The frontend prototype was designed to be somewhat interactive, allowing target users to provide feedback in early usability tests.

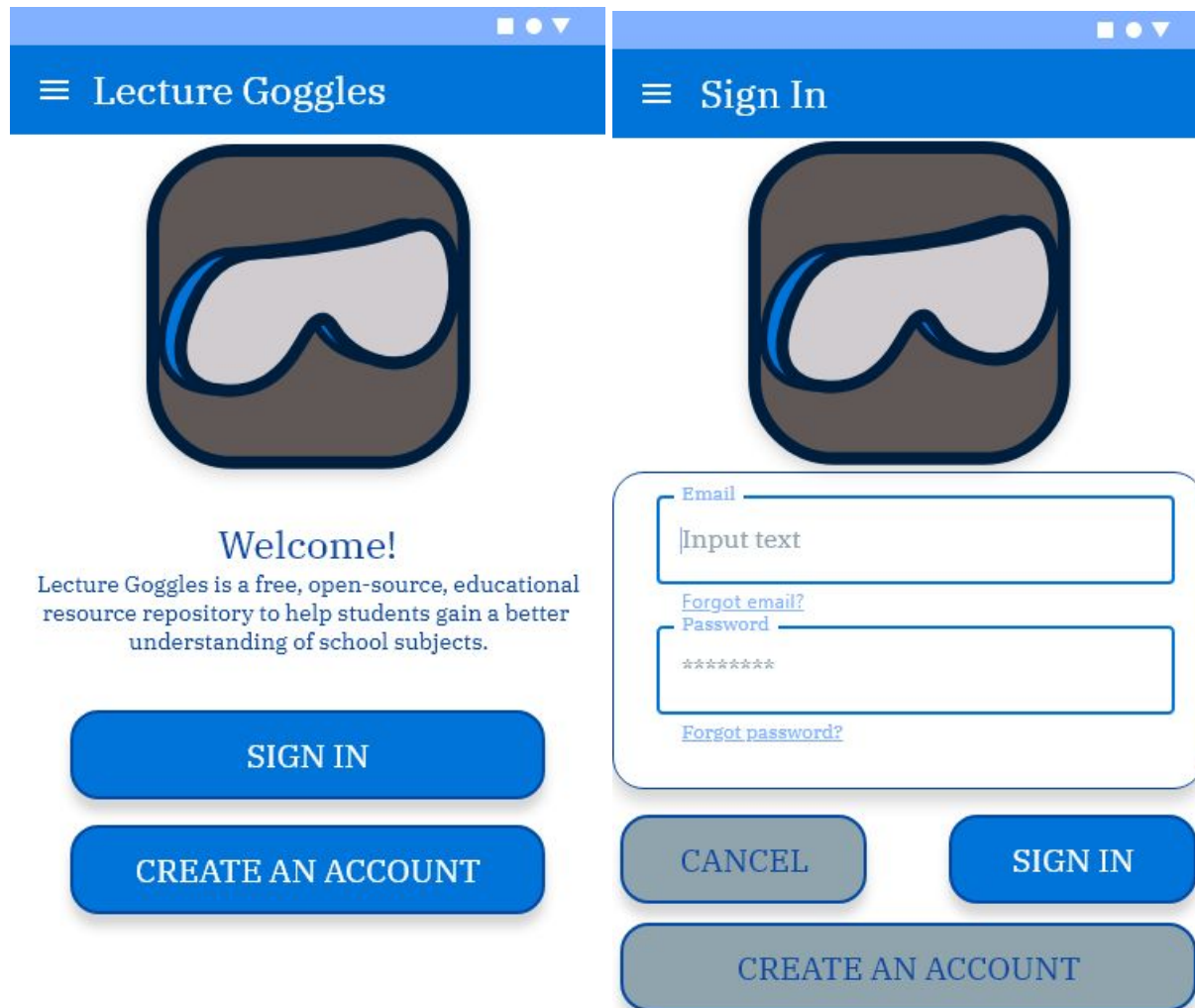


Figure 3: Home screen and sign in page on the prototype.

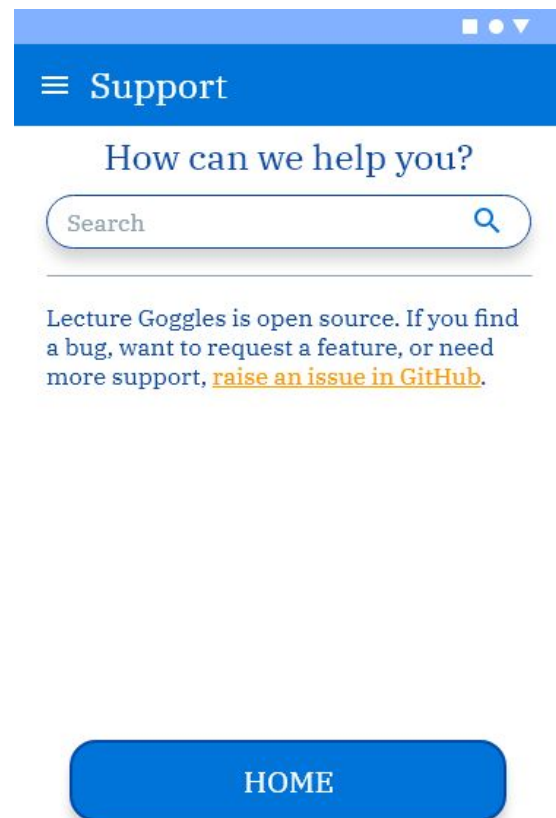
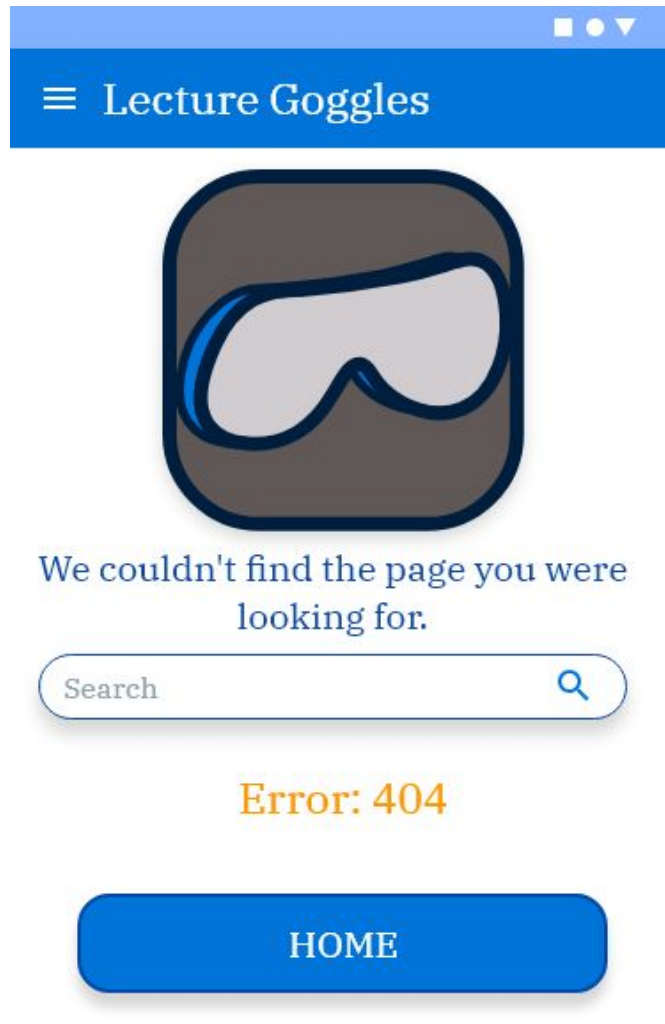



Figure 4: Not found page displayed when a link is not found on the server, and the support page in the frontend prototype.

≡ Create an Account



Lecture
Goggles

First Name

Input text

Last Name

Input text

Email

Input text

Confirm Email

Input text

Password

Confirm Password

Institution

Input Text

Are you an instructor at your institution?

☒ Yes ☐ No

CANCEL

CONTINUE

Figure 5: Account creation prototype.

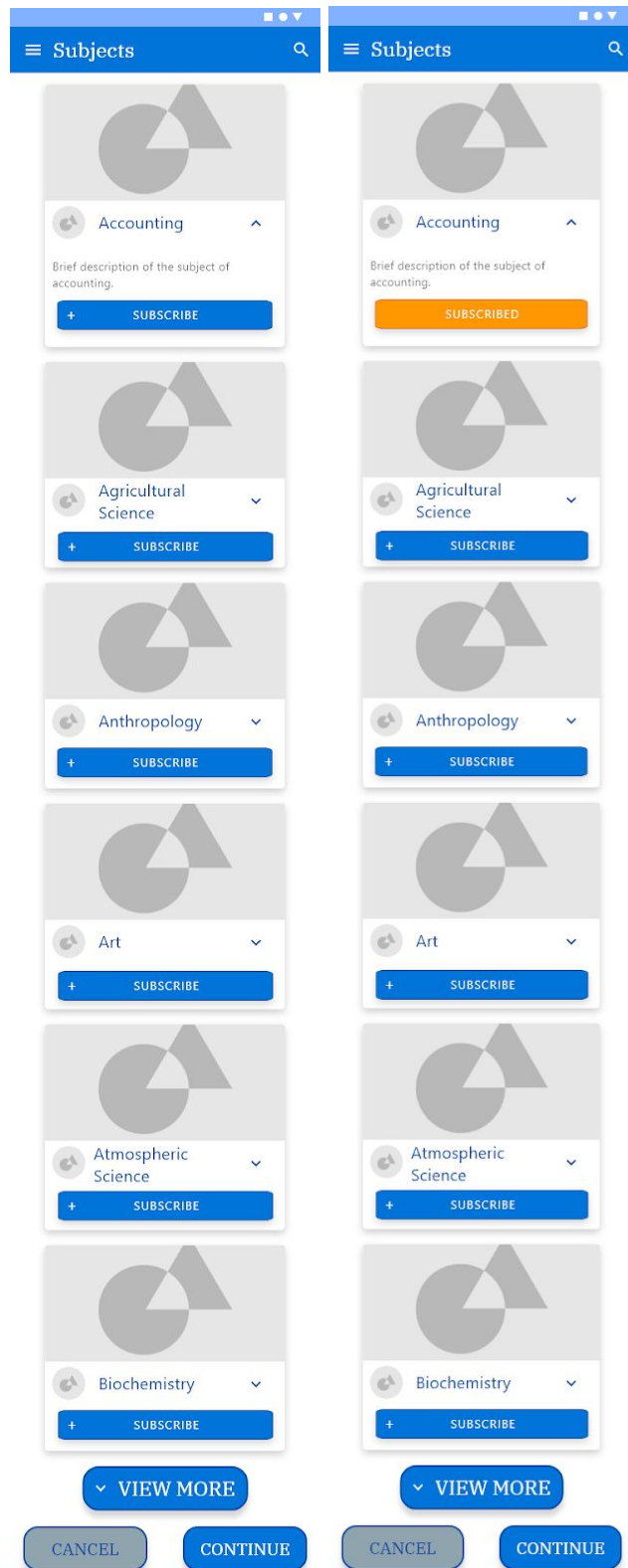


Figure 6: Browsing and subscribing to subjects.

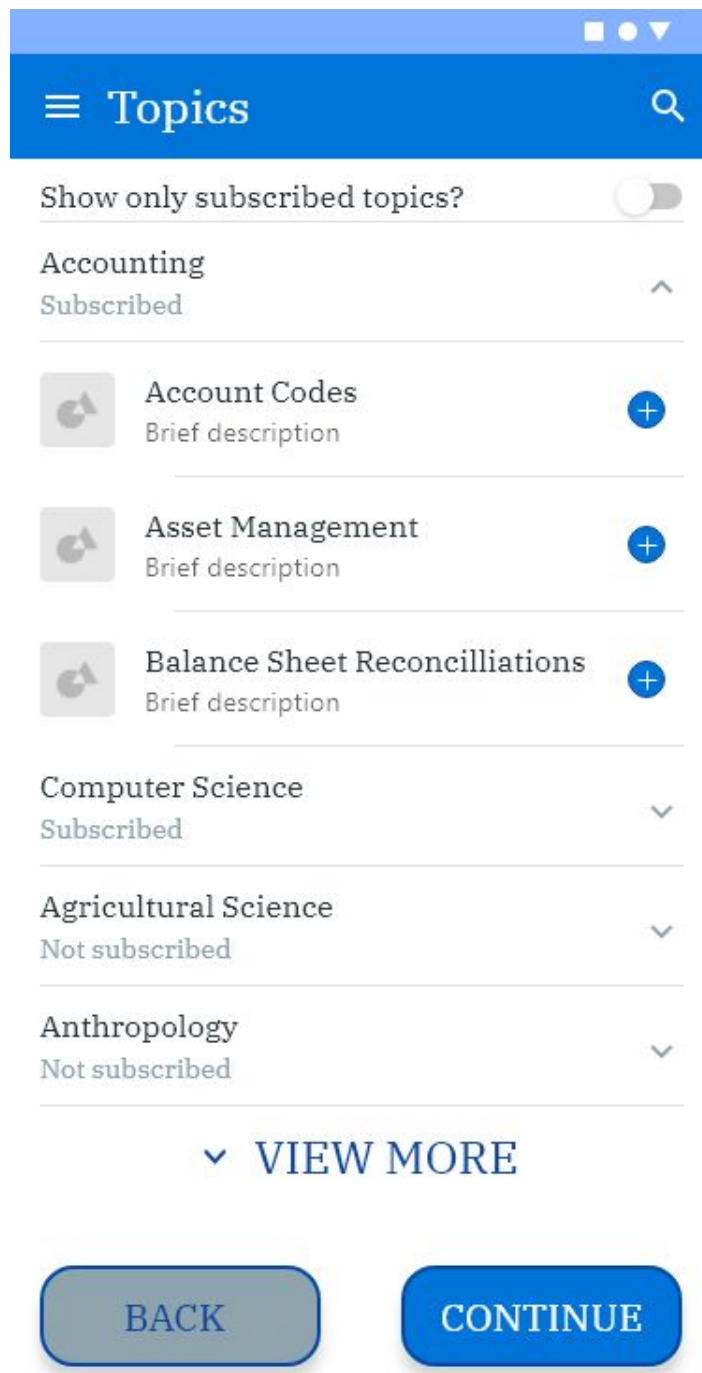


Figure 7: Browsing and subscribing to topics in the prototype.

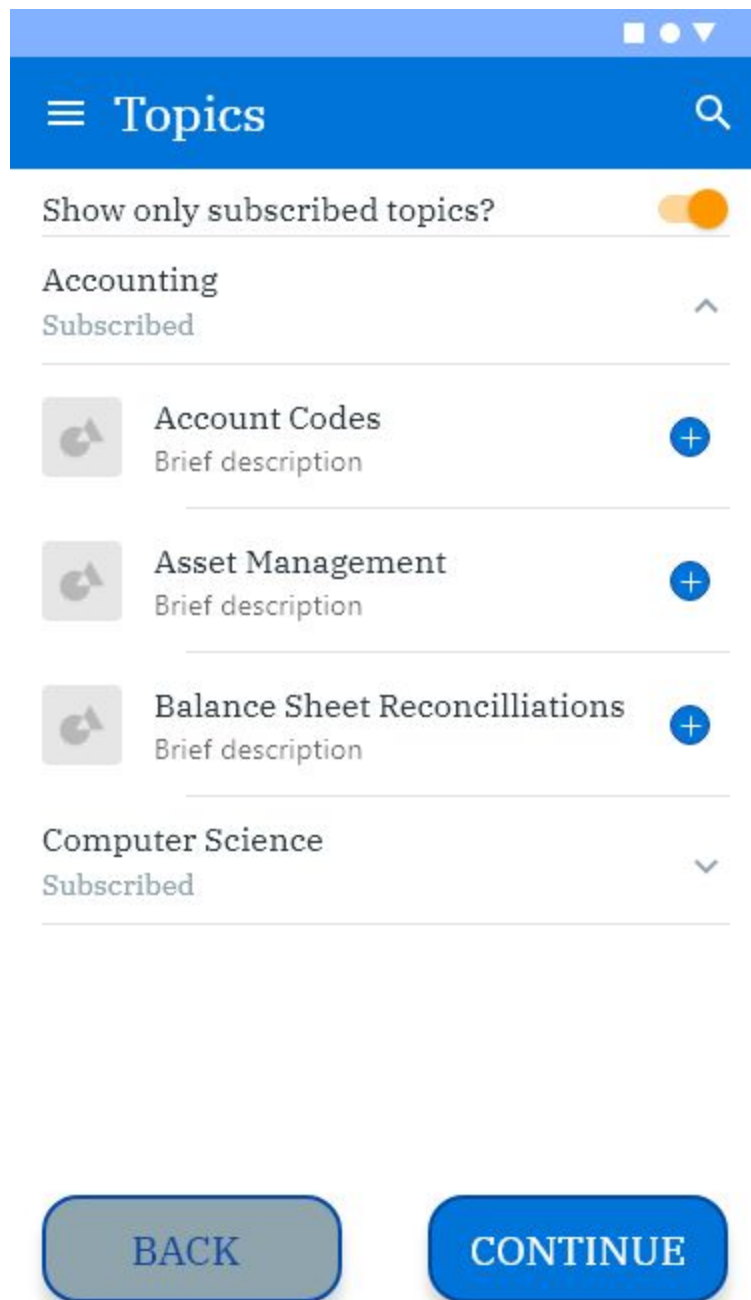


Figure 8: Filtering topics to only show subscribed topics in the prototype.

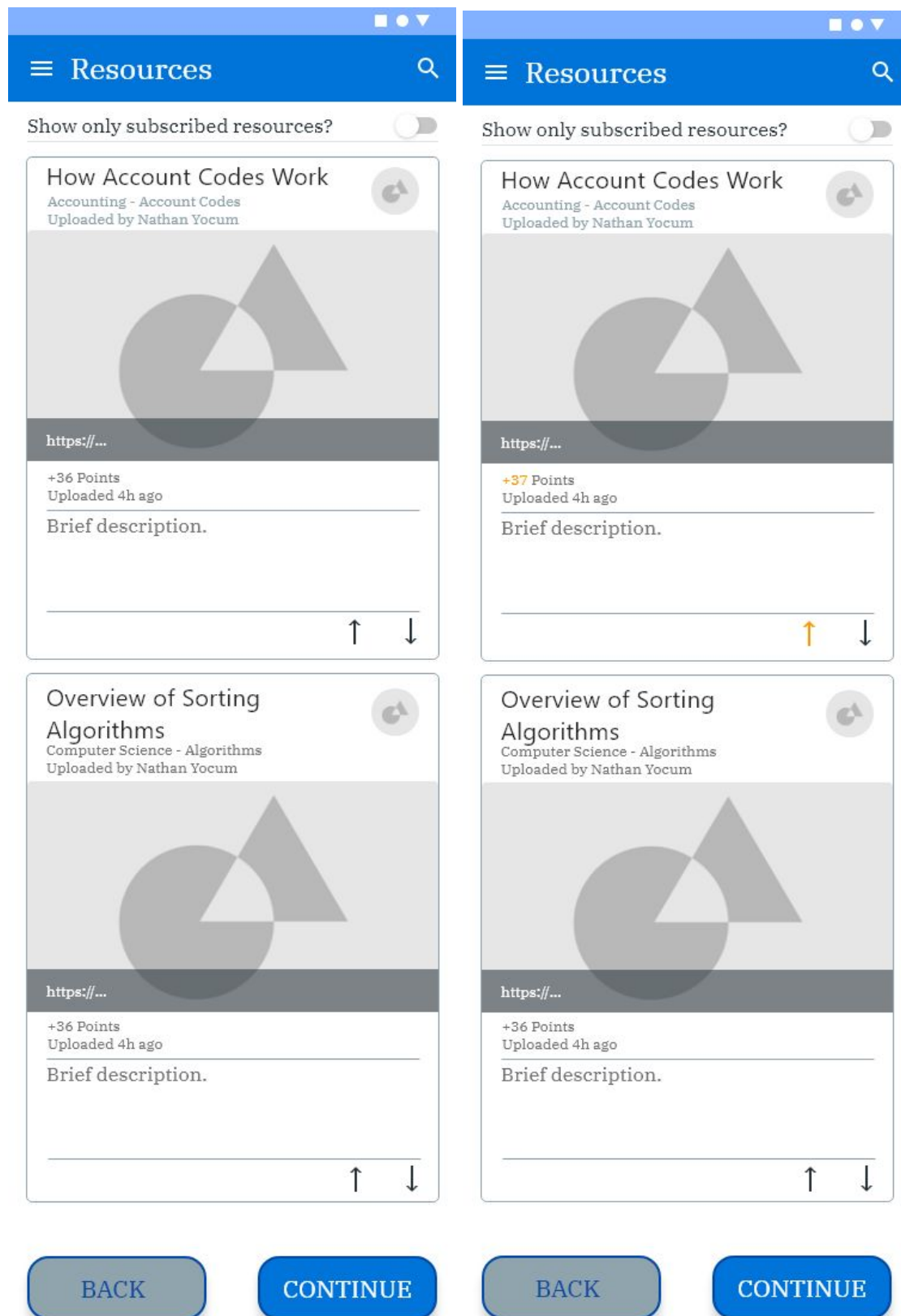


Figure 9: Browsing resources and upvoting resources in the prototype.

A rough frontend was constructed using React to test the configuration of the servers and feasibility of the design. Figure 10 displays an example of one of the pages prototyped.

Lecture Goggles

Subjects Topics Resources Support Developers Sign In Create an Account

Lecture Goggles

First Name
Last Name
Email
Confirm Email
Password
Confirm Password
Institution
Are you an instructor at your institution?
☐ Yes ☐ No

Continue
Cancel

Lecture Goggles

Subjects Topics Resources Support Developers Sign In Create an Account

Developers

Web API

Lecture Goggles has a API based on REST principles, which developers can utilize to create more applications to help students learn. Requests to the API can get, post, put, and request to delete resources.

Getting an API account

Any registered user with a verified email can request an API key. Follow [this](#) link to register for a key.

Read the documentation

...

API Methods

...

Sample applications

[Resource Randomizer](#)
...

Source Code

Lecture Goggles is a fully open source applicaion. Read about how to contribute to the [frontend](#) or the [backend](#) of the website. Lecture Goggles is licensed under an [MIT](#) licence.

Figure 10: Frontend prototype in the browser. Developer page and the sign in page.

Evaluate Prototype

The meeting with a stakeholder happened on Wednesday, December 12 at 10:30 p.m. The stakeholder was asked to use the prototyped front end. The stakeholder pointed out some minor spelling mistakes, such as the “Create Account” page saying “Confirm email” instead of “Confirm password”. She said that the interface seemed intuitive and that she liked the icons.

She commented and said that she would use the service. She mentioned that it seemed to be hard to locate specific topics, which is something we will address in the full product. She had not used the product before and was unfamiliar with our topic. She was commenting on how useful the idea was while using the front end prototype.

After explaining the idea, the stakeholder added some comments about features she would like to see. She listed the following:

1. Login with other social media accounts (Facebook, Google, etc.)
2. Full desktop website
3. Filter in search

The login options are currently not possible with the method we plan to use to validate users, but may be possible to implement when the service is expanded outside of the university level.

The full desktop site is in full development, however, the Adobe XD prototype only demonstrates the mobile version of the sight.

The filter is something we plan to add as well but is not currently being developed.

Demo Prototype

The demo team meeting for Team 22 was December 11, 2018, at 2:30 pm. Our meeting went very well with limited feedback from the instructors. They said that at some point in the future we may need to limit our project scope but could decide on that on our own. The team having Shamik Sengupta could help us have one-on-one meetings with education professionals to receive more feedback.

Changes Needed to Software Design

During the professor evaluation and current implementation of the project Team 22 was recommended to change little. Current changes are minimal, with small environmental changes such as operating system choices for server hosting. Currently, Team 22 has built a skeleton of the main components of the project. Each component can work relatively well with the other. This may change once more work has been completed in the Spring semester. The user interface will need to be redesigned to create a better navigation experience, per the prototype evaluation.

Contributions of Team Members

Zachary Johnson did the Abstract, Introduction, Django REST API portion of the Develop Prototype, Demo Prototype, and Changes Needed to Software Design 2 hours.

Logan Long did the Prototype Objectives & Functionality, and Evaluate Prototype. Logan worked on the project for approximately 2 hours.

Nathan Yocum worked on the Prototype Objectives & Functionality, the Frontend portion of the Develop prototype section, and the Changes Needed to the Software Design. Nathan worked on the prototype for around 16 hours and on the report for around 2 hours.

All three members were involved with checking each other's work and removing errors.