WWU Honors Program Senior Projects

WWU Graduate and Undergraduate Scholarship

Spring 2020

NimbleKnow User Documentation

Camille Estee Ottaway
Western Washington University

Follow this and additional works at: https://cedar.wwu.edu/wwu_honors

Part of the Computer Sciences Commons, and the Higher Education Commons

Honors Capstone Project

User Documentation

Nimble Know Classroom Feedback System

Author:

Camille Ottaway

Advisor:

Dr. Qiang Hao

Academic year 2019-2020

CONTENTS

1

INTRODUCTION

1.1 MOTIVATION

It is very common for students to become disengaged during lectures, especially on college campuses with lectures of over 100 students. Studies have shown that students learn better when engaged and asked to solve problems[1][2]. A real-time classroom feedback system can be expensive and are typically more accessible to some students than others. Students without their own devices, who speak another language, or who are blind, are excluded by these technologies.

1.2 DESCRIPTION

Nimble Know is a simple web application that teachers can use to pose questions which students can then answer using their basic smartphones or tablets. In order to accommodate ESL learners our application includes translation features. Having a user-friendly classroom technology can promote more engagement and collaboration between students and faculty alike in a classroom environment.

1.3 PROJECT DEVELOPMENT APPROACH

The Agile software development methodology was employed for this project. This methodology required our team to set goals that are can be accomplished within two-week long sprints during the development phase. During these sprints, weekly meetings between the team members were held. These 2 meetings allowed every person involved with this project to discuss progression and confirm their level of understanding of individually assigned tasks.

1.4   Division of Work

The NimbleKnow development team consisted of Camille Ottaway, Julianne Sem, Nick Stanton, and Grant Chou. Each team member had a different role within the project. Camille, the project lead, pitched the idea for Nimble Know and created the application and database design. She implemented the file system, teacher and student authentication, the ability to create questions and responses, and the top bar website navigation. Additonally, she wrote the User Documentation (this document) and hosted the website the the public can access it. Julianne did the visual design and html bootstrap coding to create the aesthetics of the website. Nick implemented the translation feature and create the home and about pages. Grant implemented the visualization feature, wrote initial unit tests, and helped with the initial question functionality. Team members worked together on ongoing design decisions and planning.

1.5   Features

The primary feature of this product is a website providing an interactive interface where teachers can easily pose questions that students can answer via their own devices.

The secondary features are as follows:

- Ability for teachers to create and log in to accounts

- Dashboard where teachers can view classroom and questions

- Teachers can create multiple choice or long response question

- Ability for students to answer questions

- Visualizations for multiple choice responses

- Integrated text translation

1.6   Similar Technologies

There are other technologies available that allow presenters to ask questions a collect audience responses, but the all have some limitations in design, or a paywall. Poll everywhere is designed for corporate settings and is poll and analysis focused. The aesthetics

are basic, and it is not designed for classroom use[4]. Mentimeter is another tool designed for corporate use. It allows people to create interactive presentations. This technology has a paywall [3] Socrative, which is designed for classrooms, provides easy, user-friendly access for students and teachers, but only allows teachers one question per day with the free version[5].

2

For Consumers

2.1 Website Consumers

Website consumers are teachers and students. This includes university professors and students as well as k-12 teachers and students. Further, NimbleKnow is flexible enough to be used in less traditional settings, including distance learning, community education opportunities such as CPR classes, and enrichment learning. The website is designed to be intuitive and we expect most end consumers will need no additional documentation to use the product. The site top-bar and well-labeled buttons makes it easy to navigate.

2.2 Accessing the Website

The website can be accessed using the embedded link here: KnimbleKnow. The full url is: http://nimbleknow.herokuapp.com. The site can be accessed through most browsers including Safari, Firefox, Internet Explorer, or Chrome.

2.3 For Teachers

2.3.1 Registering

First time users who wish to pose questions must first first register. The website home page has a large centered button that says "Teacher Register" or teachers can use the drop down

login menu. Teachers must register with a unique email address and a password, with which they later can use to login.

2.3.2   Understanding the Dashboard

The teacher dashboard for logged-in teachers displays a list of the classes they have created. To create a question, teachers must first make a class using a classname and a simple password. When teachers select a class, they can then create new multiple choice or short response questions for that class, or view responses to other questions in that class.

2.3.3   Supporting Students Use

Teachers should provide students with the url to Nimble Know and the instructions below. Note that students do not have individual accounts, they simply long into the classroom using classroom credentials.

2.4   For Students

2.4.1   Accessing Classroom

Follow the steps to access your classroom:

1. Open http://nimbleknow.herokuapp.com in a web browser

2. Select "Student Login"

3. Enter the Class Name your teacher provided

4. Enter the password your teacher provided

5. Click "Log In"

2.4.2   Answering Questions

1. Select a question

2. Answer the question

3. Click "Submit"
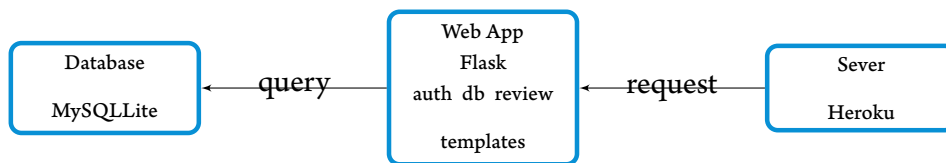
2.4.3   TRANSLATION

1. Click "Select Language"

2. Select the language you want to translate into in the drop-down menu

3

FOR PROJECT DEVELOPERS

3.1   DESIGN PLAN/ARCHITECTURE

3.2   TECHNOLOGIES USED

3.2.1   FLASK

We chose to use Flask to develop our web app. Flask is an easy to use web development framework that uses python. It was perfect for this project because it was relatively easy to learn, used python, and could develop many different kinds of projects.

3.2.2   HEROKU

Our site is hosted on Heroku at https://nimbleknow.herokuapp.com/. Heroku allowed us to host our website publicly for free. It was easy to use in general, but setting up the site could be tedious in places. Overall, we found Heroku to be a great tool for hosting our site.

3.2.3   SQLITE

We used SQLite to create and manage our database. Our database held account information as well as the classrooms associated with the accounts, the questions associated with the class-

rooms, and the responses associated with the questions. SQLite was mildly difficult to learn and use, but we found it easier than many other options.

3.2.4 CHART.JS

We used chart.js for the visualizations of our response data. Chart.js is an open-source tool used for creating HTML5 based JavaScript charts. While we weren't able to implement live updates in our charts, we were able to refresh the page and our charts would update based on our new collected data. We found chart.js to be a very easy tool to use.

3.3 FILE STRUCTURE

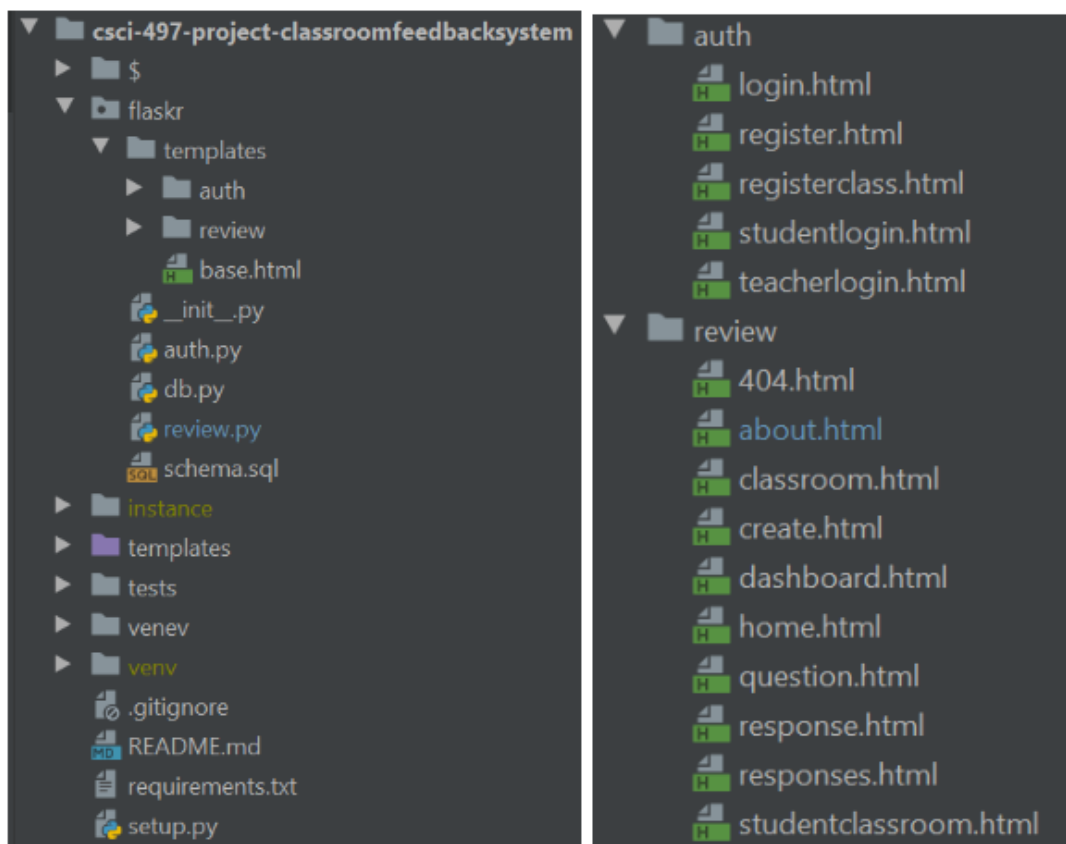Figure 1: File Structure

The above photo shows the file structure of our project. The bulk of our application is in the flaskr folder. This contains __init__.py, auth.py, db.py, review.py, and schema.sql, which are the main files that run the webapp and create and update the database. Inside this folder is

a templaes folder (right) that contains a base html template, plus auth templates and review templates. The test folder contains python files with unit tests. Most of our other files server other purposes, such as the procfile for hosting on Heroku and our requirements.txt.

3.4 Setting up the Development Environment

3.4.1 clone the repository

$ git clone git@github.com:wwu-csci-497/csci-497-project-classroomfeedbacksystem.git
$ cd csci-497-project-classroomfeedbacksystem

3.4.2 Create and activate a virtual environment

$ python3 -m venv venv .
$ .venv/bin/activate
    Or on Windows cmd:
$ py -3 -m venv venv
$ venv

3.4.3 Install Flaskr

$ pip install e
Or if you are using the master branch, install Flask from source before installing Flaskr:

    $ pip install -e ../..
$ pip install -e .

3.4.4 Run

$ export FLASK$_A$PP $= flaskr$
$exportFLASK_ENV = development$
$flaskinit - db$
$flaskrun$

    Or on Windows cmd:
$ set FLASK$_A$PP $= flaskr$

$setFLASK_ENV = development$
$flaskinit − db$
$flaskrun$

3.4.5 VIEW

Open http://127.0.0.1:5000 in a browser.

4

DEVELOPMENT DISCUSSION

4.1 CHALLENGES

There were a few challenges we faced during development, including our unfamiliarity with the technologies we were using, our failure to incorporate Object Oriented Design, and database queries. Most of us haven't used Flask before, so it was difficult understanding how the files were organized. We also had many moving pieces under development at once and putting them together was challenging. Further, our inexperience with Full Stack Development meant we unintentionally implement our web application without incorporating object oriented design. We used Flask form framework directly with the database instead of having base classes in our code. This led to additional challenges when querying the database. Initially it was challenging to understand how to use the database with Python. A few of us haven't used SQL in a while or haven't used it before, so writing the queries was difficult, as was displaying information from the database on the website.

4.2 LIMITATIONS

One major limitation our application has is that it does not have complete usability features to be used widely. The other major limitation is that improving these features and scaling

application for widespread use would require it to become more complicated, or require an overhaul transition to object oriented design.

4.3 Areas of Improvement

4.3.1 Testing

Quality testing could improve NimbleKnow. Performing user testing would give an idea of what features are important, and how easy the website is to use, and ideas for how to improve the site to make it more usable. Additionally, we only prepared unit testing during the first half of our development phase, so further implementation of unit tests would be ideal to ensure that the software is robust.

4.3.2 Convert to Object Oriented Design

Due do the current design of the application, it is complex, hard to change, and potentially prone to error. Converting the code base to object oriented design would solve all of these issues, and be worth the considerable amount of time it may take.

4.3.3 Usability

There are several additional features that if implemented, would contribute to the usability of the website. These include the functionality for teachers to edit and delete questions and classrooms. Additionally, it would be great to include password reset and forgot username functionality, as well as the ability to delete teacher accounts.

4.3.4 Question Diversification

More question features would make it possible for teachers to use the website for different type of class content. We would like to support additional question types, including true or false and fill-in-the-blank questions. Additionally, teachers may like to upload images as parts of questions or answers. Finally, more complex visualizations, such as word maps, would give teachers a better understanding of student learning for long response questions.

4.3.5 Text to Speech

Embedded text to speech services would increase the accessibility of the website, similar to our incorporation of translation services. This is something we began implementing but were unable to finish.

LICENSING

This product is licensed under the MIT License, as defined as such:

MIT License

Copyright (c) 2019 Camille Ottaway, Julianne Sem, Nick Stanon, and Grant Chou

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

REFERENCES

[1] Peat, Mary and Franklin, Sue *Supporting student learning: the use of computer–based formative assessment modules.* British Journal of Educational Technology

[2] Nicol, David J and Macfarlane-Dick, Debra, *Formative assessment and self-regulated learning: A model and seven principles of good feedback practice*, Studies in higher education

[3] *Mentimeter*, https://www.mentimeter.com/

[4] *Poll Everywhere*, https://www.polleverywhere.com/

[5] *Socrative*, https://socrative.com/

[6] *Flask*, https://flask.palletsprojects.com/en/1.1.x/