PuStack – A Supplementary Learning App

Aakanksha Ardhapurkar (ardhapu2) – Moderator: Michael Tyler Splitt
Neha Sharma (nsharma5) – Moderator: Ajay Shekar

1. Abstract:

1.1 Project Purpose

A platform for students in India to access supplementary material for their syllabus. Since India follows different school syllabi (CBSE and ICSE) there are very few platforms that have learning material. While websites such as Udemy, or Khan Academy cater to some similar topics, we wanted to create a platform that students can use for grades 9-12, and a platform that schools can use to customize learning material for better learning comprehension.

1.2 Background/Motivation

This project is a continuation of a project which we did last semester. While we developed a web app last semester, this semester we wanted to try mobile development and develop the app for iOS. This would also give us the opportunity to learn Swift, which we have been wanting to for a while. While the web app turned out okay – we want to pursue this later on as an app we can market – we realized that allowing users to access the learning material when they're on the go would cater to a wider audience.

2. Technical Specifications:

2.1 **Platform:** iOS
2.2 **Programming Languages**: Swift (takes care of both front-end and back-end).
2.3 **Stylistic Convention:** proper commenting, spacing, and naming conventions according to Swift – found here: https://github.com/github/swift-style-guide
2.4 **SDK:** iOS 10
2.5 **IDE**: XCode 8, Playground, Sublime, CloudKit, Sketch, Balsamiq
2.6 **Tools/Interfaces**: iPhone, Chrome
2.7 **Target Audience:** Students in India wanting to learn by using supplementary material.

These are just some of the technical aspects we will be focusing on to create a holistic Swift app. Most of our learnings will come from this source: https://designcode.io/. This book provides tutorials and lessons on everything from design to development.

3. Functional Specifications:

3.1 Features

Most of the features we plan to include in the mobile app have been migrated from the web app, to keep the consistency. Obviously, if we feel that we need to add more features because we need transition screens or simply because they were missing, we

will add them on during the design or development stage. The features we plan to include are:

1. Sign Up/Sign In – We will store the user details after the first time a user creates an account.
2. Users can explore the different subjects offered (History, Civics, Computer Science, etc.) for each grade. Users can also click on each topic to get a more comprehensive view of the videos offered.
3. Users can enroll in topic(s) which are then saved under 'Saved Courses' which they can access from the navbar. The 'Saved Courses' helps save time for users so that they can directly access their courses instead of searching for it again and again.
4. Once they've enrolled in a course, they can access all the videos offered by clicking on the video name.
5. This opens up a new screen where they can toggle through the videos by either clicking on 'Next' or 'Previous' or through another navbar we provide for easier navigating.
6. On the web app, we allowed users to download a zip file of the videos and learning materials, however we aren't sure if we can implement this for a mobile app.
7. Users can also access their account settings and change information such as email and password.

3.2 Scope of Project

The project is limited to these basic features currently. We aren't allowing users to customize their own syllabus as of yet. We also don't allow Facebook logins as of now.

4. Timeline:

4.1 – Setting up database and wireframing for iOS

- Neha
    - Set up XCode and any dependencies and/or packages we might need
    - Create query documentation (API)
    - Create the database using SQLite (importing all youtube videos which we have).
- Aakanksha
    - Create model documentation
    - Designing ER diagram to represent database
    - Create wireframes first using Balsamiq and then Storyboards/Sketch

4.2 – Start front-end development and finalize server API

- Neha
    -
    - Start implementing basic transitions to go from each screen to another
    - Start filling in models
    - Create view controller scaffolds needed
    - Start testing interactions and front-end elements/write manual test plan
    -
- Aakanksha
    - Create minor animations for screens
    - Start filling in models

- Implement API request methods in Swift
- Set up unit testing/mock testing environment for back-end in iOS

4.3 – Design complete front-end of app

- Neha
  - Link requests and models to user interface
  - Finish models in Swift
  - Implement enroll feature, enabling users to store subjects they're interested in for easier access.
  - Finish testing backend (API requests)
- Aakanksha
  - Finishing linking requests and models to user interface on pages requiring videos
  - Tweak any pages with videos to take care of different phone orientations and sizes
  - Finish models in Swift
  - Finish any animations left on screens

4.4 – Finish developing all features, animations and rigorously test complete app

- Neha
  - Finishing linking requests and models to user interface on pages requiring videos
  - Enable sign up/login screens allowing users to store information.
  - Finish rigorous testing suite to ensure proper debugging
  - Debug app on iPhone to catch any minor bugs
- Aakanksha
  - Implement accounts page, allowing users to change and save new data such as emails and passwords.
  - Finish implementing second tree-view navigation bar enabling toggling of videos
  - Debug app on iPhone to catch any minor bugs

5. Future Enhancements

Since we are most likely continuing to work on this app, we would definitely want to extend this app onto Android phones. We would also like to enable different kinds of log-ins (such as Facebook). We would also like to add more customizable options for users, or providing suggestions of other topics they can enroll in.