LearnIt – Project Report

Prepared by

Chirayu Joshi (18IT042)

Under the supervision of

Prof. Nehal Patel &

Prof. Ravi Patel

Abstract

learnIt is a platform which is designed for students across the globle to learn any technology. learnIt allows anyone to share thier knowledge globally. learnIt is available as mobile app as well as webapp. learnIt provides one out of the box feature that currently no large scale online education platform provides which is live streaming. Through live streaming, one teacher can teach multiple students in real time from distant places.

Project Overview

On the web platform of learnIt, the website contains features like watching live streams, getting live stream key from server, upload videos,pdf,images,etc , security by login/logout. Details of each and every feature and how to use it on webapp or mobile app is described below:

1) Watch live streams

To watch live streams of streamers on learnIt, visit **home page** by clicking on LearnIt present in navigation bar. You can find multiple streams on this page with **streamer name** and **live** tag on thumbnail of live video. You can also live streams on mobile browser. You can also watch it in fullscreen.

2) Go Live

If you want to share your precious knowledge with other learners on this platform, you can freely do that by getting streaming key from server. You can do so by clicking on **Go Live** present in navigation bar. After getting streaming key, open OBS Studio. And inside OBS Studio -> Settings -> Stream -> StreamKey, paste the streaming key provided by server.

3) Upload your course

If you want to teach someone, but who is not available at that time, you can upload your course and the learners will watch it at their pace. You can upload your course by clicking on **Upload** and then choosing your audio/video/text/image files. The files are

automatically renamed in order they are uploaded. You will receive success or failure response once you click on Upload File button.

4) Learn from other's course (in Beta version)

If you want to learn other's course, you will find their playlist in **Courses** link along with their names.

5) Mobile App

Now a days, majority of students prefer to learn from mobile phones as they can carry their mobile phones anywhere and can learn with just a single tap. So, LearnIt is also available as a mobile application. It has similar navigation and interface as web to keep it simple for users.

Technical Details

WebApp:

To create modern and dynamic webapplication, I used the currently latest, most in-demand and cutting edge web technology **ReactJS**. Using **axios** package of ReactJS I sent request to server and fetched response which contained live streams, streaming key, login credentials, course files, etc. I used **Video.js** to display live stream which is in MPEG-4 format. I used **react-router-dom** package for routing in client side. For upload progress, I used **react-toastify** and **reactstrap** for displaying upload progress.

To create secure, robust and highly flexible server which can handle multiple request at a time and live stream feature, I used **NodeJS**. I implemented Login/Logout feature by adding **passport.js** package in NodeJS. By passport.js, I created sessions using which user don't have to login each time he visits learnIt. The main focus of learnIt i.e Live Streaming is implemented using **Node Media Server.** When user streams using their media encoders like OBS Studio, etc. our server receives stream in the form of flv (flash) format in **RTMP** (Real Time Messaging Protocol) protocol. RTSP (Real Time Streaming Protocol) is not used as it not so popular. Flash format was supported in older browsers, but now it's use has been depricated. So, our Node Media Server converts flv stream into **HLS** (Http Live Stream) format by using **FFMPEG** (Fast Forward MPEG) library of linux. By using HLS stream, even mobile browser can play live video easily, but as downside, it increases load on server. I used MongoDB database because NodeJS (particularly **Express.js**) supports MongoDB well which increases query speed. I used **fs** and **multer** package of NodeJS to store uploaded files into server after updating their names.

Fork me on Github:

https://github.com/chirayu-joshi/learnIt