A VIDEO NOTE TAKING SYSTEM TO MAKE ONLINE VIDEO LEARNING EASIER

Haochen Han[1] and Yu Sun[2]

[1]Haochen Han, Troy High School, Fullerton, CA 92831
[2]California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

Recent coronavirus lockdowns have had a significant impact on how students study. As states shut down schools, millions of students are now required to study at home with pre-recorded videos. This, however, proves challenging, as teachers have no way of knowing whether or not students are paying attention to the videos, and students may be easily distracted from important parts of the videos. Currently, there is virtually no research and development of applications revolving specifically around the subject of effectively taking digital notes from videos. This paper introduces the web application we developed for streamlined, video-focused auto-schematic note-taking. We applied our application to school-related video lectures and conducted a qualitative evaluation of the approach. The results show that the tools increase productivity when taking notes from a video, and are more effective and informational than conventional paper notes.

KEYWORDS

Web Service, Note Taking, React JS.

1. INTRODUCTION

Note-taking is one of the most important factors in facilitating students' learning [1, 2, 3]. To be competent learners, students must be able to capture key information within fast-moving lectures [4, 5]. Traditionally, this was done by pen and paper. The recent popularization of online courses and video learning has presented a new challenge to learners worldwide, namely, how to take effective notes in a fast manner. There are existing note taking apps, but most of them are not taking advantage of the digital nature of online learning, and are merely an online version of pen and paper notes (where students type info into a digital document). ***There are two missed opportunities for existing note taking apps***. ***First***, online videos are a brand-new learning format compared to traditional lectures in terms of their recordability and repeatability. Thus, a good note-taking app should be able to take advantage of online videos. However, this feature is not present in most modern note-taking apps. Parrotnote, however, does take advantage of the video format of online courses and streamlines the note-taking experience for online learning. ***Second***, traditional note taking applications require users to painstakingly format their notes, potentially taking away valuable learning time and distracting them from the content of their lessons. This manual formatting can also be chaotic and messy, as users might not understand what is the best format for taking notes. Thus, one may consider auto-formatting a natural solution for such issues. Taking away users' freedom to edit in exchange for organization, effectiveness, and a more traditional formatting is what sets Parrotnote apart from other note-taking applications. We believe this is the right design choice because a learner's priority is to have a format that allows written information to be easily memorized without wasting time struggling with a custom editor

DOI: 10.5121/csit.2021.110917

(e.g., Evernote). The goal of Parrotnote is to improve the speed of note-taking and thus make video learning more effective, accessible, and convenient.

One example of a digital note taking system that has been proposed to help improve learning effectiveness is Evernote [6, 7, 8]. However, Evernote does not take full advantage of the digital nature of content and is still confined by some of the limitations of traditional documents, such as requiring students to painstakingly format their notes and constantly worry about the readability and structure of their information. This slows the learning progress, and even worse, many students may not know what note-taking format is best for them, thus creating messy notes in which information gets scrambled and becomes hard to recognize [9, 10]. Parrotnote recognizes such issues and incorporates a new approach to note-taking in response to them—full automation with limited customization. Instead of presenting the user with a blank document sheet, our application provides a clear, concise, and effective user interface that lets the user enter info. On the other hand, a more traditional approach to taking video notes would be writing down info on either a google doc or a piece of paper. These methods, however, have their own downsides beyond the formatting issues mentioned previously. Taking notes on a google doc or paper requires constantly closing or pausing the video, switching tabs, and typing new info. This can easily interrupt the concentration of the user, and potentially distract them while switching tabs [11]. Also, such methods do not take advantage of the digital nature of video. Parrotnote, however, does not require either pausing video or switching tabs. Our user simply needs to enter a link to their video. Parrotnote will then automatically fetch the video source from the web and display it inside the app, where the student can then take notes. This eliminates the chore of tab switching and helps the user stay focused. One of the most prominent developments in recent years on the subject of note-taking is the Cornell Note Format [12, 13, 14, 15]. Parrotnote's auto-format generation system is essentially designed to follow such a format to produce maximum clarity and readability.

Our tool to solve the problem of online digital note taking is a website that allows users to simply type notes while watching videos without worrying about formatting. The main advantage of this method is that students can have a fluid experience without being overly distracted from the video. Also, students will not have to worry about the formatting of their notes, which makes our tool convenient and simple to use. The core feature behind our tool is the ability to generate a digital, Cornell note formatted pdf based on the information the user types while watching the video. Users simply have to enter the video lecture link on the website for an embedded video to show up. Beside the video panel is the note panel, in which the user can type individual pieces of information. Each piece of information is indexed and time stamped, and later on is used to generate a PDF note in Cornell Note Format.

The effectiveness of our web note taking application is much greater than that of the traditional switching tabs approach or the traditional pencil and paper approach, so long as the user is familiar with how to use a computer and a website. As our first step, we conducted research on user experience, asking several users to test the effectiveness of the website. The results were mostly positive, as many users reported that they enjoyed the convenience of the note taking process, as facilitated by the website. Several other users found the different slots for different types of notes quite useful (our website has two tabs in the note taking section, one for information and one for questions). This shows that our website is user friendly. Secondly, we studied the behavior of less than a dozen users. We found that users were less likely to be worried about the format of their notes while using our website (because our website could format notes automatically) and could instead focus on grasping the key concepts of the lecture videos. Because they could focus more on the video and less on the actual formatting of the notes, their note taking speed and effectiveness greatly improved.

The rest of this paper is organized into different sections. First, Section 2 will list three challenges that we faced while implementing our ideas. The section following will explain the core features and implementation details.

2. CHALLENGES

In order to create a video note taking system to make online video learning easier, a few challenges were identified, as follows.

2.1. Challenge 1: Generation of the PDF Note

One challenge that we faced is the actual generation of the PDF note. The core idea of Parrotnote is to create a user-friendly experience of taking notes from a lecture video, as video lectures are likely to be prevalent during quarantine. We adopted a two-step solution and implemented a backend server. First, the frontend website will send all the note data to our backend server through a HTTP GET request. The server uses mustache.js, a templating language, to render a HTML string from the data. Afterward, the HTML string is sent to the frontend web as the response and is generated into a file through Javascript BLOB object.

2.2. Challenge 2: Embedding Videos into Parrotnote

Another major feature that posed a great challenge was embedding videos into Parrotnote. Parrotnote allows users to grab their favorite lecture videos from YouTube, then play them on the Parrotnote webpage while allowing notetaking through the notetaking features. To implement browser video, we had three options: 1. YouTube Embedded Video (Iframe tag in HTML), 2. saving the video on our server first, then streaming it to the user through a custom video streaming, and 3. Letting the user upload videos to the webpage. We decided to use YouTube Embedded Video, or the Iframe tag, simply because it would be easier for users to pick videos from YouTube rather than uploading them from files. There is an enter field in the application prompting users to enter their video links. After getting the video links, Parrotnote automatically generates YouTube Iframe Embedded links by grabbing the unique YouTube video IDs from the videos. We use a Regular Expression statement to grab the IDs.

2.3. Challenge 3: Implementing A Secure, Reliable Server and Providing the Note Generation Feature

There was also the challenge of implementing a secure, reliable server to actually host our website and provide the note generation feature mentioned in the first challenge. We chose not to do server side rendering, and instead used React.js to build a single page application. This is because server side rendering usually results in more overhead on the server, which is something we did not want. By reducing the overhead of the server side rendering, we cut the work of the server down to essentially two functions: serving the website, and providing the formatting note GET route. Another step we took to make our server more robust was to cloud host it on an AWS server, which provides faster internet speed.

3. SOLUTION

Parrotnote is an online note taking web application that formats note segments such that users can enter notes digitally while watching lecture videos on the same tab. While using the website, users will first be prompted to enter a YouTube Link to the lecture video they want to study. Afterward, an embedded video will appear on the right side of the screen, which users will watch.

On the left side are the note sections. Users can enter pieces of information or questions there, and the note sections will automatically be numbered. When users finish taking notes they can click on the PDF button to get their notes generated in Cornell format, and then get the html page for their notes. There are three main subsystems. The first two are at the frontend, and they comprise the system for grabbing YouTube video embedded codes from YouTube links and the system for saving note data from user input. These were coded in react.js. Finally, there is a backend server, which will open an API GET route, which receives users' note data, and generates html page strings to send to the frontend.

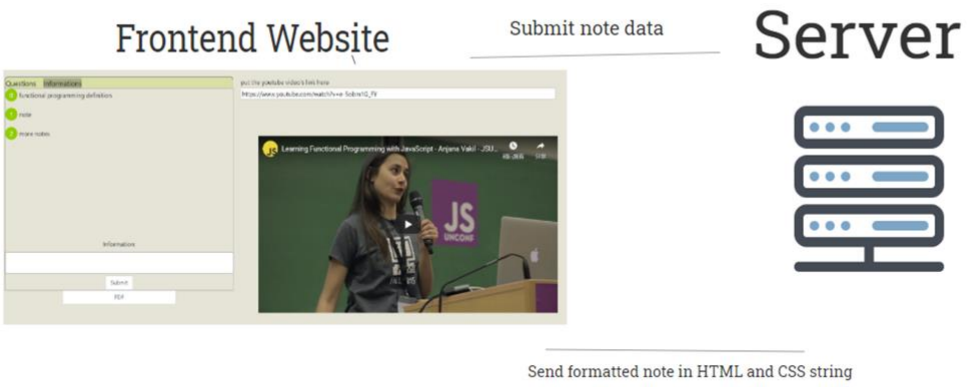
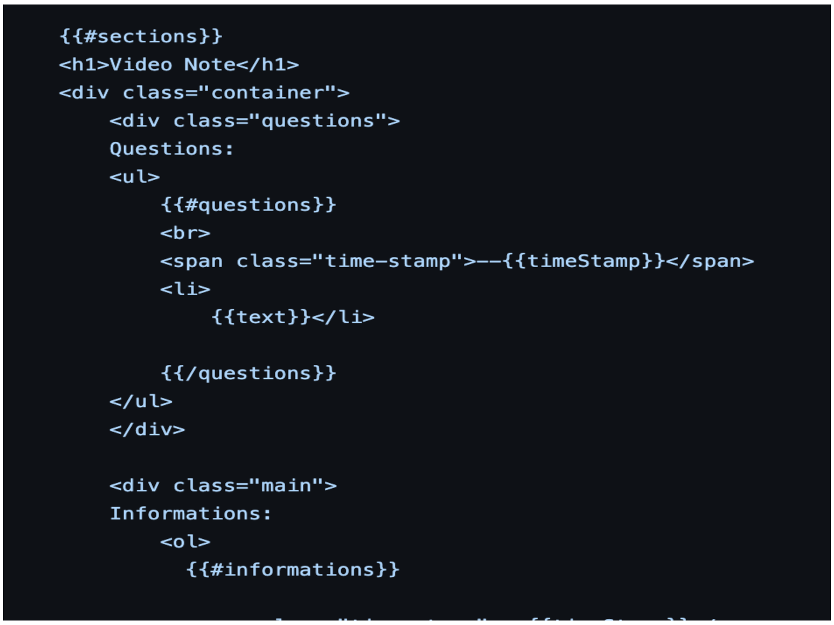

Frontend Website
Submit note data
Server
Questions
Informations
functional programming definition
note
more notes
put the youtube video's link here
https://www.youtube.com/watch?v=e-5obm1G_FY
Learning Functional Programming with JavaScript - Anjana Vakil - JSU...
Information
Submit
PDF
Send formatted note in HTML and CSS string

Figure 1. Parrotnote system

Backend

We used node.js as the primary language for the backend. For routing and regular items, we simply used express to handle them. The core feature of the backend is the ability to turn user submitted note data into pdf notes, save them, and serve them to users. This feature was achieved with templating language and css styling. In essence, we used html and css to first create the style for the pdf, then used Mustache.js templating language to fill data in its place.



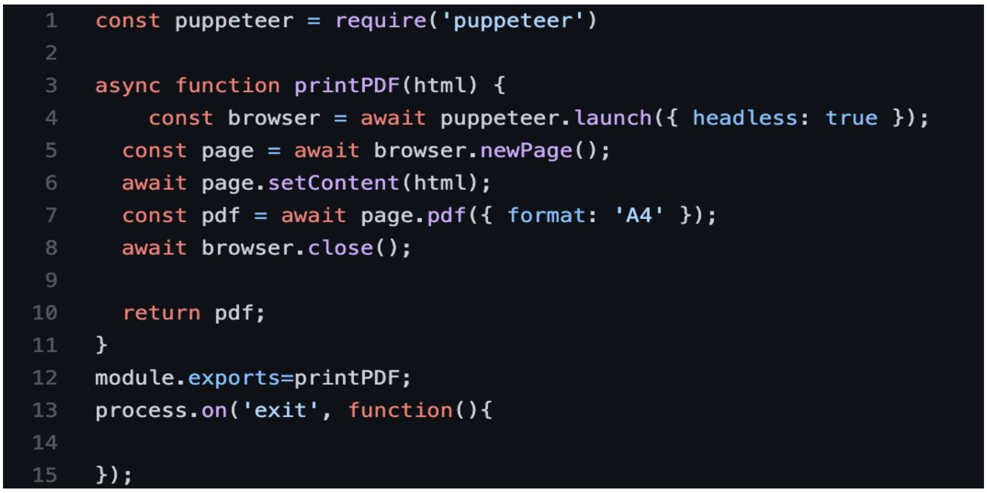
```
{{#sections}}
<h1>Video Note</h1>
<div class="container">
    <div class="questions">
    Questions:
    <ul>
        {{#questions}}
        <br>
        <span class="time-stamp">--{{timeStamp}}</span>
        <li>
            {{text}}</li>

        {{/questions}}
    </ul>
    </div>

    <div class="main">
    Informations:
        <ol>
          {{#informations}}
```

Figure 2. Html formatting

However, one tricky issue we faced with html formatting was actually turning this into a pdf, as users may not need html files. There are many techniques to achieve this. We used a headless browser to pre-convert html to pdf before sending to the user.



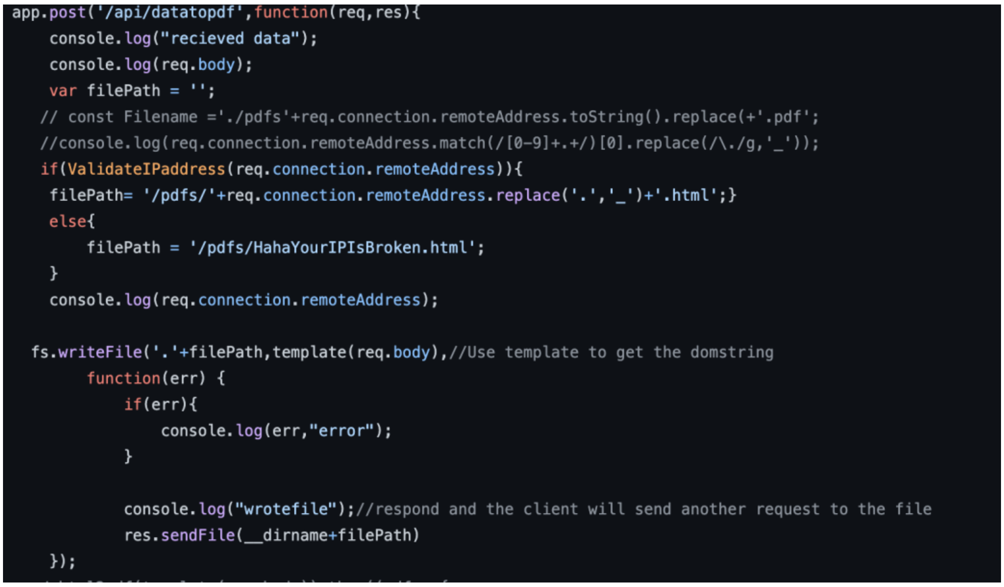
```
const puppeteer = require('puppeteer')

async function printPDF(html) {
    const browser = await puppeteer.launch({ headless: true });
  const page = await browser.newPage();
  await page.setContent(html);
  const pdf = await page.pdf({ format: 'A4' });
  await browser.close();

  return pdf;
}
module.exports=printPDF;
process.on('exit', function(){

});
```

Figure 3. Code



```
app.post('/api/datatopdf',function(req,res){
    console.log("recieved data");
    console.log(req.body);
    var filePath = '';
   // const Filename ='./pdfs'+req.connection.remoteAddress.toString().replace(+'.pdf';
   //console.log(req.connection.remoteAddress.match(/[0-9]+.+/)[0].replace(/\./g,'_'));
   if(ValidateIPaddress(req.connection.remoteAddress)){
    filePath= '/pdfs/'+req.connection.remoteAddress.replace('.','_')+'.html';}
    else{
        filePath = '/pdfs/HahaYourIPIsBroken.html';
    }
    console.log(req.connection.remoteAddress);

  fs.writeFile('.'+filePath,template(req.body),//Use template to get the domstring
        function(err) {
            if(err){
                console.log(err,"error");
            }

            console.log("wrotefile");//respond and the client will send another request to the file
            res.sendFile(__dirname+filePath)
    });
```

Figure 4. Here we generate the file with a unique URL based on the user IP. Afterward, the user is redirected to the file and they can download it from their browser.

Frontend

The most important feature of our application is the system that allows users submit note data to our server and receive a HTML and CSS string for formatted notes, which the frontend eventually converts to BLOB, and downloads. The technologies we used include frontend javascript fetch and BLOB. In the backend, where the server renders a CSS formatted HTML page, we used Mustache.js as the rendering engine.

The second most important feature, note recording and data saving, was accomplished through javascript and react.js. We simply made a styled list. The program then objectifies each piece of note, and displays its data through the styled list.

To enable users to watch lecture videos on our website, we used Iframe technology and YouTube embedded videos. First, users will input the link to their video. Then, our program will use a Regular Expression to extract the unique YouTube identifier from each video (contained in query string). Lastly, our program will convert that unique identifier to an HTML iframe tag, based on YouTube video embedding iframe format.

4. EXPERIMENT

To test the effectiveness of Parrotnote, we conducted several user studies involving taking video notes. After dividing our testers into experimental and control groups, we randomly selected a YouTube video on programming in C# (https://youtu.be/5MmhSfyO3kg), and asked them to take notes using different techniques. Specifically, the experimental group was asked to use the Parrotnote app player to play a video and take notes using the Parrotnote editor, while the control group was asked to open a blank google doc and take notes using it. We summarized the resulting data derived from their notes and took the mean to create the following chart:

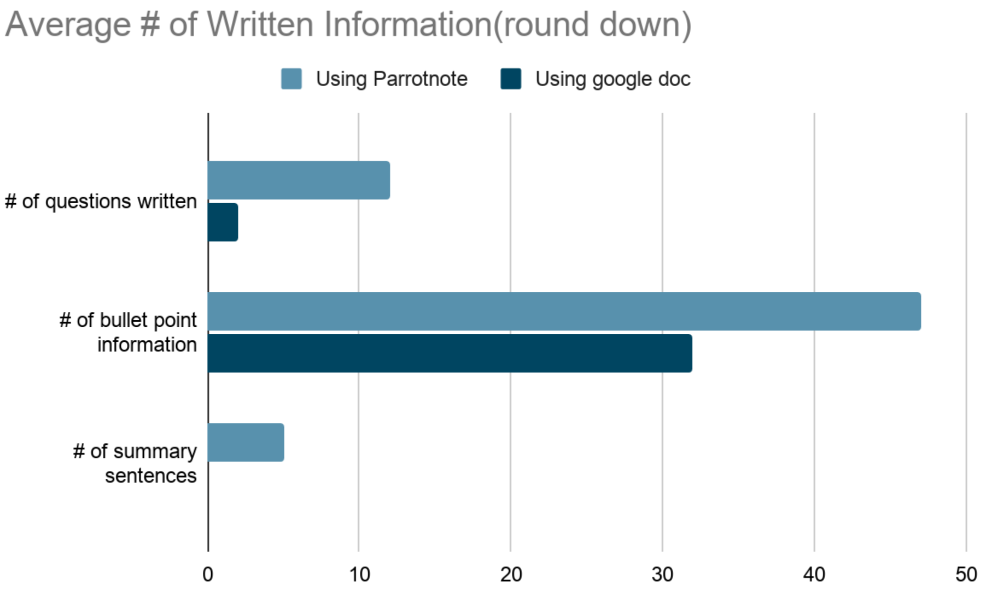


Figure 5. Chart of note-taking data

When it came to bullet point information, the experimental group (using Parrotnote) had a slightly higher number of items written down. This could likely be due to the convenient, embedded in-app video player Parrotnote offers, as it allows users to type information while watching videos without the need of pausing or switching tabs. More noticeably, Parrotnote users had a significantly higher number of both questions and summaries written down in their notes. This is likely due to the fact that Parrotnote's note editor provides a clear place for summaries and questions, thus reminding users to keep track of such information. It is also likely that when manually formatting notes using google docs, the control group users could simply get lazy and skip questions and summaries, whereas the experimental group users wrote them down because it was easier and faster to record them with Parrotnote (due to its auto-formatting feature). As the

Cornell note study points out (see also related work), summaries and questions are crucial components to help students learn.

5. RELATED WORK

Jenni Donohoo at Cornell University proposed the technique known as Cornell note [16]. It was devised to help students organize their lecture notes. This paper introduces a method by which teachers can teach students about how to make good Cornell notes. One key aspect of it is a gradual process of teaching students how to write down important information and delete irrelevant information. The strategy we employed is different. Since Parrotnote focuses on speed, students are able to write down multiple bits of information to use as review materials.

Korzaan and Lawrence have written on how Evernote pioneered the field of online, document-based note taking and cloud storage [17]. Evernote uses a document interface that allows users to format notes themselves. Parrotnote differs from Evernote in that Parrotnote automatically formats notes that users input into Cornell Note Format, instead of letting users format their notes manually.

Leann Mischel has written on the efficiency of EDpuzzle within online learning situations involving videos [18]. Our work differs from EDpuzzle in terms of method. EDpuzzle is an application that enforces video completion by locking down the browser, which may be suitable for a classroom environment. Parrotnote, on the other hand, is an application that boosts note-taking speed for self-paced learners.

6. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a new way of taking notes online while watching video lectures. While there are many tools such as EDpuzzle that are suitable for classroom use, Parrotnote is suitable for self-paced learners. We have proposed an auto-formatting approach to online note taking, which will greatly boost the speed of users when combined with embedded video players and potential speech-to-text notes. Furthermore, we have conducted experiments on our auto-formatting approach, comparing people who manually format with people who use Parrotnote. The result shows that students who use Parrotnote to help them perform auto-formatting are more likely to write down key information, such as questions and summaries, while students who type on a blank document will likely only write down plain information. This shows that our auto-formatting approach helps students gain a better and a more critical understanding of their videos. The increased speed and amount of information written down also suggests that Parrotnote can be a great tool to create review sheets or study guides. In essence, Parrotnote is a tool suitable for individual learners who seek to boost their note-taking speed and general understanding of key concepts while watching lecture videos.

In this paper, we also introduced the techniques we used, including aspects from both backend and frontend. In the backend, the key feature is the pdf note generation from data, in which we apply templating language and a headless browser. In the frontend, we described the way we present the information, and how we attempted to make it intuitive. We also introduced our embedded video player feature.

Parrotnote can currently record text-based notes. It provides a division of question notes and informational notes. There are still many limitations to the current Parrotnote application. For example, the formatting does not allow customization. It is also currently limited to text-based notes.

In future, we would devise a better system that allows for greater customization of the existing note pieces. Such a system would allow for switching orders and changing font size, etc. Also, we are planning to include the styling of users' note pieces so they can choose to highlight, bold, or italicize text.

REFERENCES

[1] Boch, Francoise, and Annie Piolat. "Note taking and learning a summary of research." *Writing.* 2005.

[2] Barnett, Jerrold E., Francis J. Di Vesta, and James T. Rogozinski. "What is learned in note taking?" *Journal of Educational Psychology* 73.2 (1981): 181.

[3] Di Vesta, Francis J., and G. Susan Gray. "Listening and note taking." *Journal of educational psychology* 63.1 (1972): 8.

[4] Howe, Michael JA. "The utility of taking notes as an aid to learning." *Educational Research* 16.3 (1974): 222-227.

[5] Horney, Mark A., et al. "Exploring the effects of digital note taking on student comprehension of science texts." *Journal of Special Education Technology* 24.3 (2009): 45-61.

[6] Zhumabekova, G. B., and A. T. Kenzhebekova. "THE EFFECTIVENESS OF SMART TECHNOLOGY AS "EVERNOTE" IN FORMATION OF WRITTEN COMMUNICATION." *Абылай хан атындағы ҚазХҚжӘТУ*: 58.

[7] Ka, Jay. "Using the Mobile Application Evernote for Diagnostic Assessment to Enhance Foreign Language Proficiency." (2016).

[8] Ansari, Mohd Shoaib, and Aditya Tripathi. "An investigation of effectiveness of mobile learning apps in higher education in India." *International Journal of Information studies and libraries*2.1 (2017): 33-41.

[9] Schuh, Allen J. "Effects of an early interruption and note taking on listening accuracy and decision making in the interview." *Bulletin of the Psychonomic Society* 12.3 (1978): 242-244.

[10] Piolat, Annie, Thierry Olive, and Ronald T. Kellogg. "Cognitive effort during note taking." *Applied cognitive psychology* 19.3 (2005): 291-312.

[11] Zureick, Andrew H., et al. "The interrupted learner: How distractions during live and video lectures influence learning outcomes." *Anatomical sciences education* 11.4 (2018): 366-376.

[12] Donohoo, Jenni. "Learning how to learn: Cornell notes as an example." *Journal of Adolescent & Adult Literacy* 54.3 (2010): 224-227.

[13] Quintus, Lori, et al. "The impact of the Cornell note-taking method on students' performance in a high school family and consumer sciences class." *Journal of Family & Consumer Sciences* 30 (2012): 1.

[14] Evans, Bradley P., and Chris T. Shively. "Using the Cornell Note-Taking System Can Help Eighth Grade Students Alleviate the Impact of Interruptions While Reading at Home." *Journal of Inquiry and Action in Education* 10.1 (2019): 1-35.

[15] Susanti, Luh Eka. "CORNELL NOTE-TAKING: EFFECTIVE WAY TO ACTIVATE STUDENTS'AUTONOMOUS LEARNING." *Journal on Studies in English Language Teaching (JOSELT)* 1.2 (2020): 43-55.

[16] Donohoo, Jenni. "Learning how to learn: Cornell notes as an example." *Journal of Adolescent & Adult Literacy* 54.3 (2010): 224-227.

[17] Korzaan, Melinda, and Cameron Lawrence. "Advancing student productivity: an introduction to Evernote." *Information Systems Education Journal* 14.2 (2016): 19.

[18] Mischel, Leann J. "Watch and learn? Using EDpuzzle to enhance the use of online videos." *Management Teaching Review* 4.3 (2019): 283-289.