

A Project Report  
On  
**Progressive Web App for Note-Taking**  
BY

**Nitin Gopala Krishna Sontineni    2018A7PS0262H**

**Nivesh Duppalapudi                    2019A7PS0018H**

**Sohan Das                                2018A7PS0848H**

Under the supervision of

**Prof. Narasimha Bolloju**

**SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF  
CS F366 LAB PROJECT**



**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)  
HYDERABAD CAMPUS**

**(May 2022)**

# ACKNOWLEDGMENTS

We would like to express our special thanks and gratitude to Prof. Narasimha Bolloju for giving us this wonderful opportunity to work on this project. His guidance and input were pivotal throughout the course of this project. We were able to learn about the workflow and pitfalls of the implementation of a software system from the ground up. This experience would help us immensely in our future careers in the industry.

We would also like to thank our parents for their continued support despite their busy schedules.

**Nitin Gopala Krishna Sontineni**  
**Nivesh Duppalapudi**  
**Sohan Das**



**Birla Institute of Technology and Science-Pilani,  
Hyderabad Campus**

**Certificate**

This is to certify that the project report entitled “*Progressive Web App for Note-taking*” submitted by Mr. **Nitin Gopala Krishna Sontineni (2018A7PS0262H)**, Mr. **Nivesh Duppalapudi (2019A7PS0018H)** and Mr. **Sohan Das (2018B3A70848H)** in partial fulfillment of the requirements of the course CS F366, Lab Project Course, embodies the work done by them under my supervision and guidance.

**Date: 04-05-2022**

**(Prof. Narasimha Bolloju)**

BITS- Pilani, Hyderabad Campus

# ABSTRACT

Note-taking is one of the most common learning activities that students perform in their academic lives. Note-taking applications provide a fast, easy, and organized way for taking notes. The main advantage of note-taking applications is that notes can be shared among the students easily. Applications having all the functionalities like viewing lectures, taking notes, and giving feedback in one place will make the learning process easier.

This laboratory project aims to develop a progressive web app that supports student note-taking while viewing the lectures and also facilitates feedback taking from the students to improve the course content by the professors. MERN(MongoDB, Express, React, Node) stack is used for the development of the application. This report outlines our background work and progress made so far in the project.

# Table of Contents

<b>ACKNOWLEDGMENTS</b>	<b>2</b>
<b>ABSTRACT</b>	<b>4</b>
<b>Table of Contents</b>	<b>5</b>
<b>Introduction</b>	<b>7</b>
<b>User Personas</b>	<b>9</b>
<b>Product Backlog (User Stories)</b>	<b>10</b>
<b>Use Case Diagram</b>	<b>11</b>
<b>Use Case Descriptions</b>	<b>12</b>
View Feedback Points	12
Make Notes	13
<b>User Interface Design of the application</b>	<b>14</b>
Proposed UI flow for the students:	14
Proposed UI flow for the professors:	14
<b>Database Design</b>	<b>15</b>
Classes	15
Professor	15
Course	15
Student	15
Lecture	16
Review	16
Notes	16
Embedding	16
Referencing	16
<b>System Architecture</b>	<b>17</b>
Justification:	17
<b>Conclusion</b>	<b>18</b>
Summary of the problem and work done	18
Summary of learning/reflections on project work	18

<b>Appendices</b>	<b>19</b>
<b>A. Code Quality Report</b>	<b>19</b>
<b>B. Sample Screenshots</b>	<b>19</b>
Login and Signup pages:	19
Student dashboard (enrolled courses):	20
Selected Course Content:	21
Student's Lecture viewing page (notes):	21
Student's Lecture viewing (feedback):	22
Professor's Lecture viewing page (feedback from all students):	22

# Introduction

Note-taking helps students to capture information that they are learning in the classroom lectures and use that information typically at a later point of time, to improve their learning. The hand-written notes are still a popular method which is followed all over the world even in this digital age. Proponents of the pen-and-paper method say that it provides a tangible object that the student interacts with, and more importantly has a personal aspect attached to it. Writing with hand can help register the information faster or retain it for a longer period of time are some of the arguments. However, the traditional method does have its drawbacks. With the advent of digital document editors and readers, comes more efficient ways of not only creating digital notes, but also searching, highlighting, extracting, editing, and sharing notes which otherwise would be very difficult to achieve. The notes that two students take, even while viewing the same lecture, are likely to be different and it can be argued that it is these differences that adds to the utility of the notes. With the help of online note taking applications one can embed images, links, tables and other informational elements into their document that can maximize the usefulness, all while adding to the individuality of their notes. Digital note-taking becomes especially important when the education systems themselves seem to be going online.

Due to the COVID-19 pandemic, most institutions across the world in different sectors have switched to online mode to continue their operations while respecting the social distancing policies setup by the governments. Educational institutions are in some sense at the forefront of this switchover. In fact some say that the transition was bound to happen regardless of the pandemic. Online mode of teaching comes with its own advantages and disadvantages. Few advantages are students can attend the class from anywhere saving them the energy and time of transportation and the classes can easily be recorded. Some disadvantages are the requirement of a stable internet connection on both the student's end and the professor's end. Many times, there are video or audio issues in the lecture which are unaddressed, and also there might be some discrepancies in the content being taught. Some topics are very difficult to understand in the online mode of teaching. These problems are very difficult to deal with, as one has to mail the professor about it and wait for a reply. Ironically, all this can increase the gap between the student and the teacher in an online environment which are otherwise seen as avenues to bring people closer.

The motivation behind our web application is to tackle the problem of interactivity between the teacher and the student along with providing a note-taking facility to students, all in one platform. Students should be able to take notes and give feedback to the teacher, all while viewing the lecture itself. Since it is a digital platform, they can study at their own pace, rewind and rewatch the videos and make notes at any timestamp of the video. They can also choose if their notes are visible to other students or only to themselves. Furthermore, our app also facilitates giving feedback for a particular lecture of a particular course at a particular timestamp. This ensures that reviews are closely tied to the topics discussed in the lectures and thereby are more effective at conveying the feedback. Any problem at any point in the recording can easily be brought to the notice of the professor. Moreover, the professor can see

reviews made by all the students in one place and address the problem. Students can also give ratings along with the feedback at particular timestamps to convey if they find a certain section interesting or requiring more explanation. This is beneficial to the professor as it provides a quick glance of the student's reaction at certain points in the video lecture.

We have used the scrum methodology for developing our project. Scrum is a lightweight agile process framework used primarily for managing software development. We divided the project into 3 sprints and each sprint had its own subtasks. We used the MERN stack for developing our application.

In the first sprint, we developed the user personas, user stories, the user interface, and the use case diagrams. Then in the second sprint, we designed our database and then finished the backend APIs. We used postman to test our backend APIs. We used Node js as our runtime and express framework for the backend. MongoDB, a NoSQL database, has been used for storing documents.


In the final sprint, we finished the frontend part of the application and connected it with the backend using Axios. The frontend part was done using React. We also used sonar cloud for the code quality report.


### **Overview of the report:**

The first section is the introduction section which talks about the importance/use of the project. The next section is the user personas. User personas are archetypical users whose goals and characteristics represent the needs of a larger group of users. Next comes the user stories. A user story is an informal, general explanation of a software feature written from the perspective of the end-user. Its purpose is to articulate how a software feature will provide value to the user. Then comes the use case diagrams and descriptions. A Use case diagram can summarize the details of a system's users and their interactions with the system. The next section is the user interface which depicts the interface of student and professor. Next is the database design which has the details about the implementation of the database at the backend. Then comes the system architecture which is the conceptual model that defines the structure, behavior, and more views of a system. The last section is the conclusion section which concludes the project along with sample screenshots and code quality report.



# User Personas

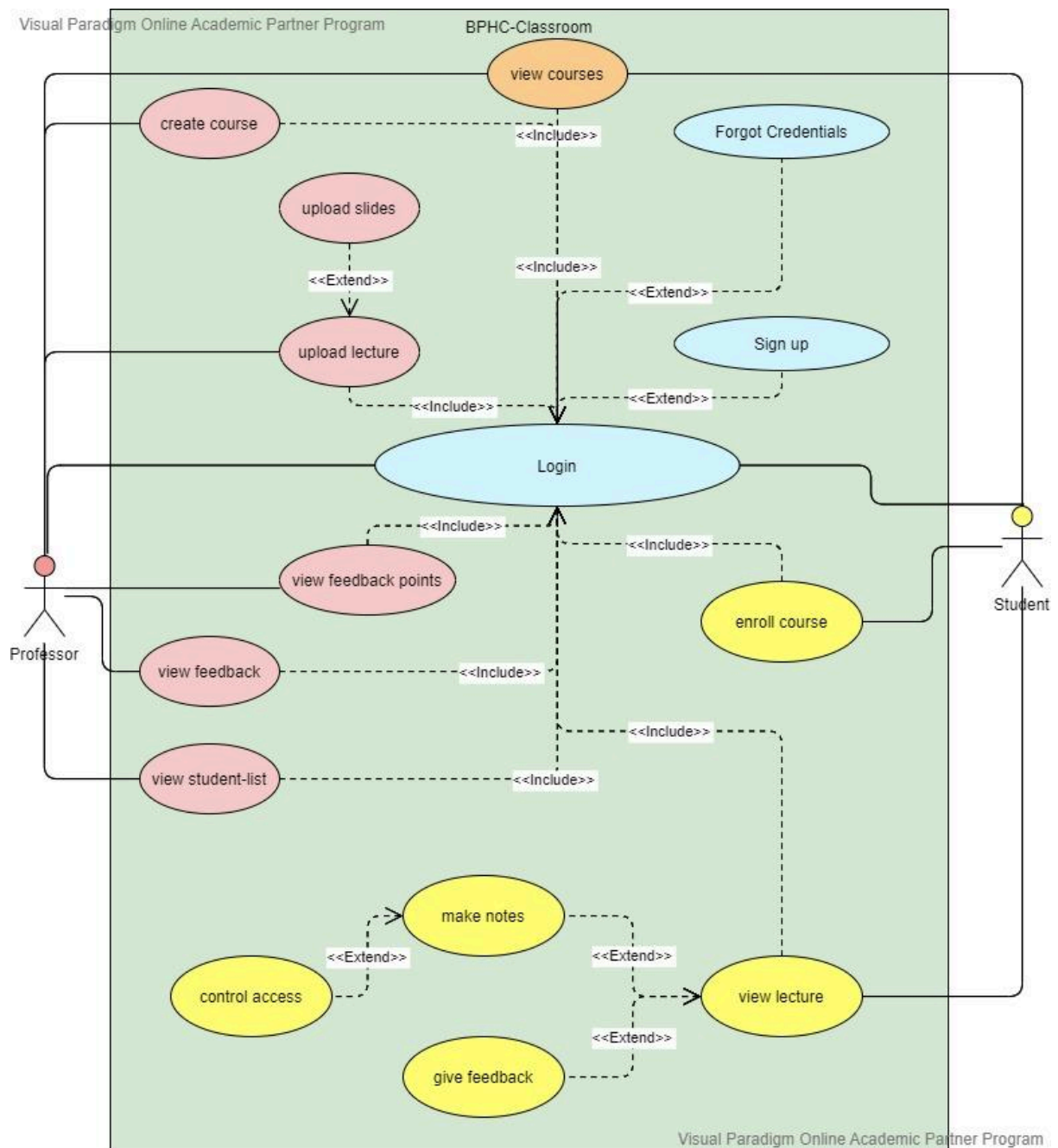
<b>Jane Doe</b> (Student)	<i>"Ability to take notes is crucial for me to understand what I am studying"</i>	
	<b>Capabilities</b> <ul style="list-style-type: none"> <li>• Good with technology</li> <li>• Makes descriptive notes</li> <li>• Was teaching assistant for one semester</li> </ul>	<b>Goals/Objectives</b> <ul style="list-style-type: none"> <li>• She wants an app which can help her to find lectures and notes all at one place</li> <li>• She also is looking for an app which would help her ask doubts regarding her topics.</li> </ul>
<b>Demographics</b> Age: 23 Education: Undergraduate Work: Student Family: Single Location: California	<b>Brief bio:</b> Jane is a physics major at UCLA. She is currently in her sophomore year and maintains a great GPA. She attributes much of her success to the ability to make easy-to-understand notes and ask doubts to her professors.	<b>Frustrations/Pain points</b> <ul style="list-style-type: none"> <li>• While she is good at making notes, she is not that organized and often loses her notebooks.</li> <li>• Doesn't like to use a multitude of apps to serve her needs.</li> </ul>

<b>Tom Stewart</b> (Professor)	<i>"I like answering questions and consider a student's feedback an important part of pedagogy"</i>	
	<b>Capabilities</b> <ul style="list-style-type: none"> <li>• Very interactive with students</li> <li>• Takes into account suggestions or feedback</li> </ul>	<b>Goals/Objectives</b> <ul style="list-style-type: none"> <li>• He wants an app which can be used to deliver lecture to his students</li> <li>• He also wants the app to improve his interaction with students, by allowing them to give feedback and ask doubts.</li> <li>• The app should be user-friendly.</li> </ul>
<b>Demographics</b> Age: 65 Education: PostDoc Work: Tenured Professor Family: Married Location: California	<b>Brief bio:</b> Tom is a tenured professor of physics at UCLA. He is old school and prefers to use traditional modes of teaching. However, due to the lockdowns he has been forced to rely on online modes.	<b>Frustrations/Pain points</b> <ul style="list-style-type: none"> <li>• He is not good with technology and often struggles to manage his lectures and slides</li> <li>• Due to lockdown, he misses the interaction he had with his students.</li> </ul>

# Product Backlog (User Stories)

As a/an	I want to	So that	Use Case
<i>Professor</i>	login using my credentials	I can access system functionality according to my role	login
<i>Professor</i>	create a new course	students can enroll to my course	create course
<i>Professor</i>	upload a lecture video of a particular course	students can view it	upload lecture
<i>Professor</i>	upload the corresponding lecture slides	students can access them	upload slides
<i>Professor</i>	see the list of students enrolled for my course	I get to know who is taking my course	view student-list
<i>Professor</i>	get the feedback of the lectures at specific timestamps	I can improve the further lectures content	view feedback
<i>Professor</i>	view at which points in the video most students made comments	give special attention to/refine contents of those areas in the lecture	view feedback points
<i>Professor</i>	view all the courses I am teaching	keep track of courses and view them at a single place	view courses
<i>Student</i>	login into my account	I can view the courses	login
<i>Student</i>	enroll for a course	I can access that course content	enroll course
<i>Student</i>	have a play, pause, forward, rewind functionalities	I can view the lecture video according to my requirement	view lecture
<i>Student</i>	make notes on the platform under specific lecture	I can refer to it later	make notes
<i>Student</i>	rate and review/add comments at any point of the lecture	it helps professors to improve and also for other students to get the idea about the course content	give feedback
<i>Student</i>	control access to my notes and comments	I can share my notes publicly or keep my notes to myself	control access
<i>Student</i>	view all the enrolled courses	I can find the required course easily	view courses

# Use Case Diagram



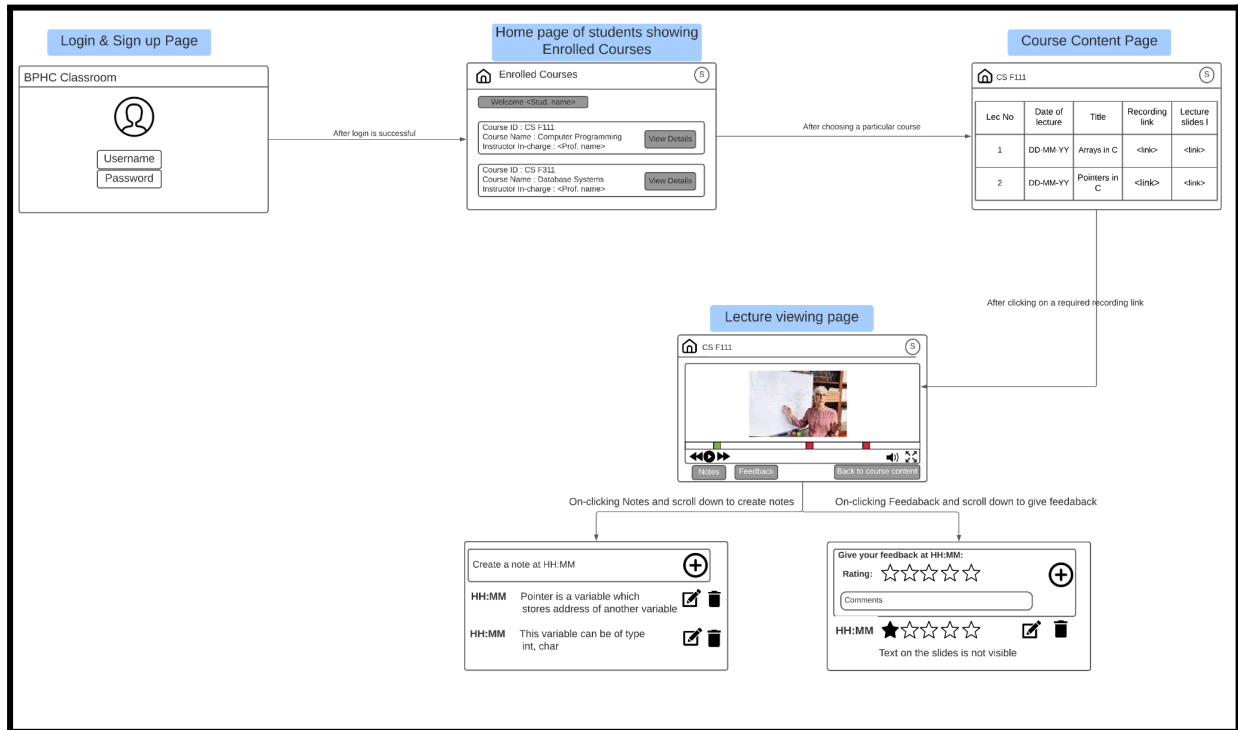
# Use Case Descriptions

<b>Name</b>	<b>View Feedback Points</b>
<b>Description</b>	Professor views red and green points on the video which signify negative and positive feedback respectively.
<b>Actors</b>	Professor
<b>Trigger</b>	Professor wants to see the feedback points on a video lecture he has uploaded.
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• Professor has logged into his account</li> <li>• Professor has selected a course he teaches</li> <li>• Professor selects one of the uploaded lectures of the particular course</li> </ul>
<b>Postconditions</b>	Professor is able to see red and green points on the video.
<b>Main course (MC)</b>	<ol style="list-style-type: none"> <li>1. Professor clicks on the lecture link of one of his uploaded lectures. (EX1, EX2)</li> <li>2. The system shows the video player and starts playing the video.</li> <li>3. Professor is able to see red and/or green feedback points on the video progress bar in the video player. (AC1, EX2)</li> </ol>
<b>Alternate courses (AC)</b>	<ol style="list-style-type: none"> <li>1. There is no feedback on the lecture currently.               <ol style="list-style-type: none"> <li>a. Go to MC2.</li> </ol> </li> </ol>
<b>Exceptions (EX)</b>	<ol style="list-style-type: none"> <li>1. The lecture link is broken:               <ol style="list-style-type: none"> <li>a. System informs the professor that the link is broken.</li> <li>b. Go to MC1.</li> </ol> </li> <li>2. Professor clicks on 'Log out':               <ol style="list-style-type: none"> <li>a. System logs out the professor account.</li> </ol> </li> </ol>

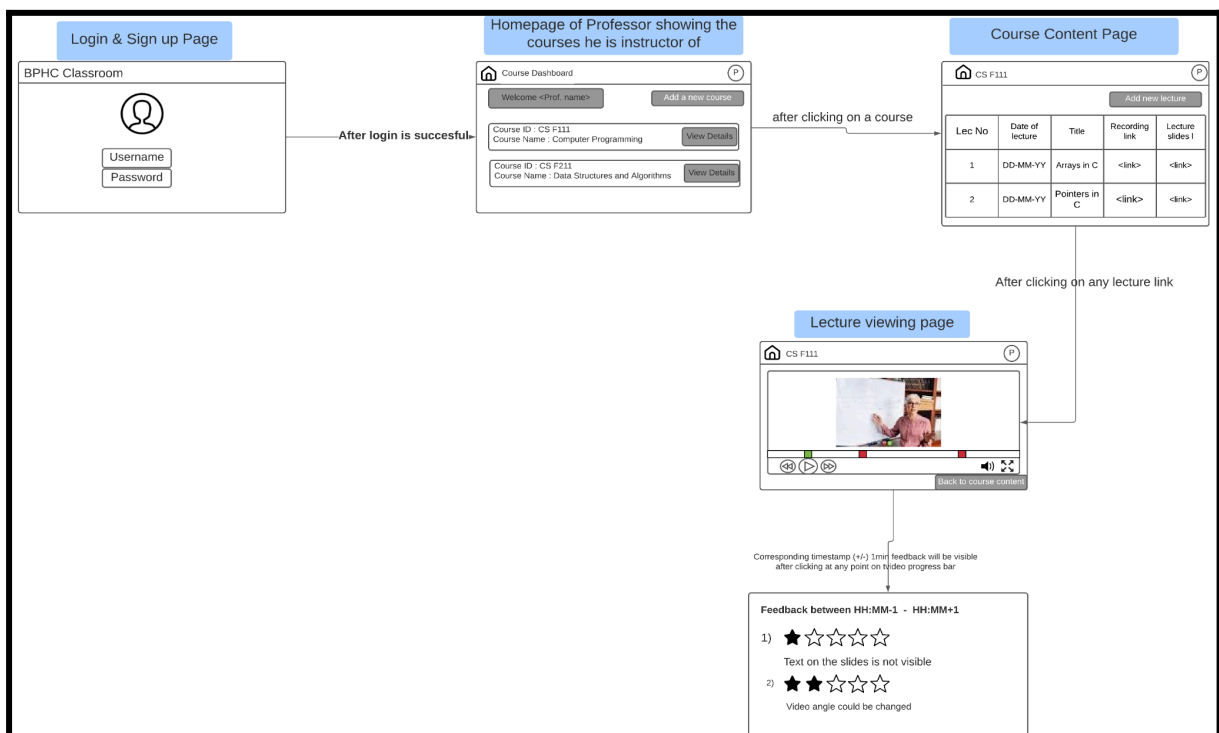
<b>Name</b>	<b>Make Notes</b>
<b>Description</b>	Student makes notes at a certain timestamp of the video lecture.
<b>Actors</b>	Student
<b>Trigger</b>	Student wants to add a note at a certain timestamp of the video.
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• Student has logged in.</li> <li>• Student has enrolled in a course.</li> </ul>
<b>Postconditions</b>	Student has added a note at a certain point of time of the video lecture.
<b>Main course (MC)</b>	<ol style="list-style-type: none"> <li>1. Student clicks on the lecture link for one of the lectures under the course she has selected. (EX1, AC1, EX3)</li> <li>2. Student clicks on the 'Note' button.(EX3)</li> <li>3. Student types her note in the text field. (EX3)</li> <li>4. Student clicks on the '+' button. (AC2)</li> <li>5. System adds the note along with the timestamp of the video when the '+' button was clicked by the student.</li> <li>6. Student views her newly added note. (EX3)</li> </ol>
<b>Alternate courses (AC)</b>	<ol style="list-style-type: none"> <li>1. Currently, there are no lectures in the enrolled course: <ol style="list-style-type: none"> <li>a. System says that there are no uploaded lectures for the selected course.</li> <li>b. Go to M1</li> </ol> </li> <li>2. Student clicks elsewhere, rather than the '+' button after typing the note in the text field: <ol style="list-style-type: none"> <li>a. System prompts the user to continue typing or discard the note (EX2)</li> <li>b. Go to MC3.</li> </ol> </li> </ol>
<b>Exceptions (EX)</b>	<ol style="list-style-type: none"> <li>1. The lecture link is broken: <ol style="list-style-type: none"> <li>a. System informs the student that the link is broken.</li> <li>b. Go to MC1.</li> </ol> </li> <li>2. Student discards note without adding: <ol style="list-style-type: none"> <li>a. System clears the text field.</li> <li>b. Go to MC3.</li> </ol> </li> <li>3. Student clicks on 'Log out' button: <ol style="list-style-type: none"> <li>a. System logs out the student account.</li> </ol> </li> </ol>

# User Interface Design of the application

## Proposed UI flow for the students:

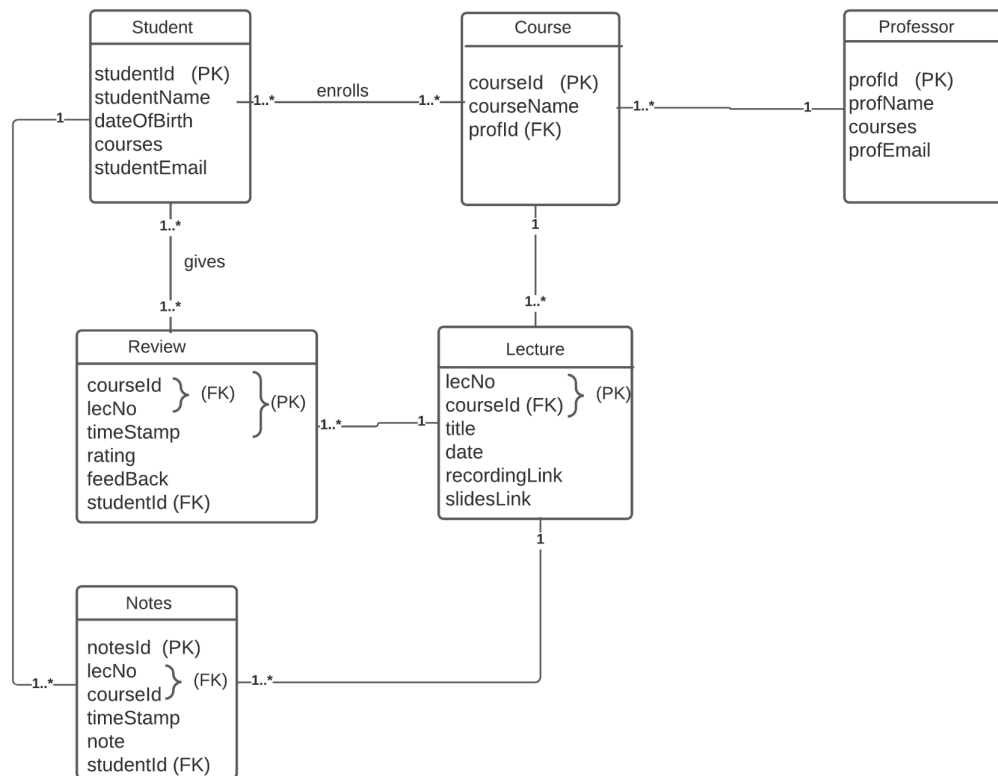


## Proposed UI flow for the professors:



# Database Design

## BPHC Classroom Database Design



## Classes

### Professor

This class contains the professor's Id (profId) , the professor's name, the courses he/she is in charge of (courses), and the professor's email. Embedding is used for courses.

### Course

This class firstly contains course's Id (courseId) which is the primary key. The next field is the name of the course (courseName). The final field contains the id of the professor who is in charge of that course (foreign key).

### Student

This class contains the basic information of the student like student name (studentName) , student id (studentId) , date of birth of the student (dateOfBirth) , courses in which the

student is enrolled (courses), and the email address of the student (studentEmail). Embedding is used for the 'courses' field.

## **Lecture**

In this class, the primary key is the combination of lecture number (lecNo) and course Id which is a foreign key. It also has other fields like the title of the lecture recording, date, recording link, and slides link.

## **Review**

In this class, the combination of courseId , lecture number, and timestamp is the primary key. It also contains the rating points (rating), feedback (feedback), and studentId which is the foreign key.

## **Notes**

This class has notesId as the primary key (notesId) . It also has fields like note which contains the notes taken by the student and the time when the note was taken (timestamp). It also contains the fields like lecture number (lecNo), course Id (courseId), and student Id (studentId) which are foreign keys.

## **Embedding**

Embedding is used in Professor (courses field) and Student (courses field) classes. Since the courses in which a student is enrolled will be accessed multiple times, embedding is a better option than referencing. We are only storing the course IDs inside the courses array so it won't take up much space.

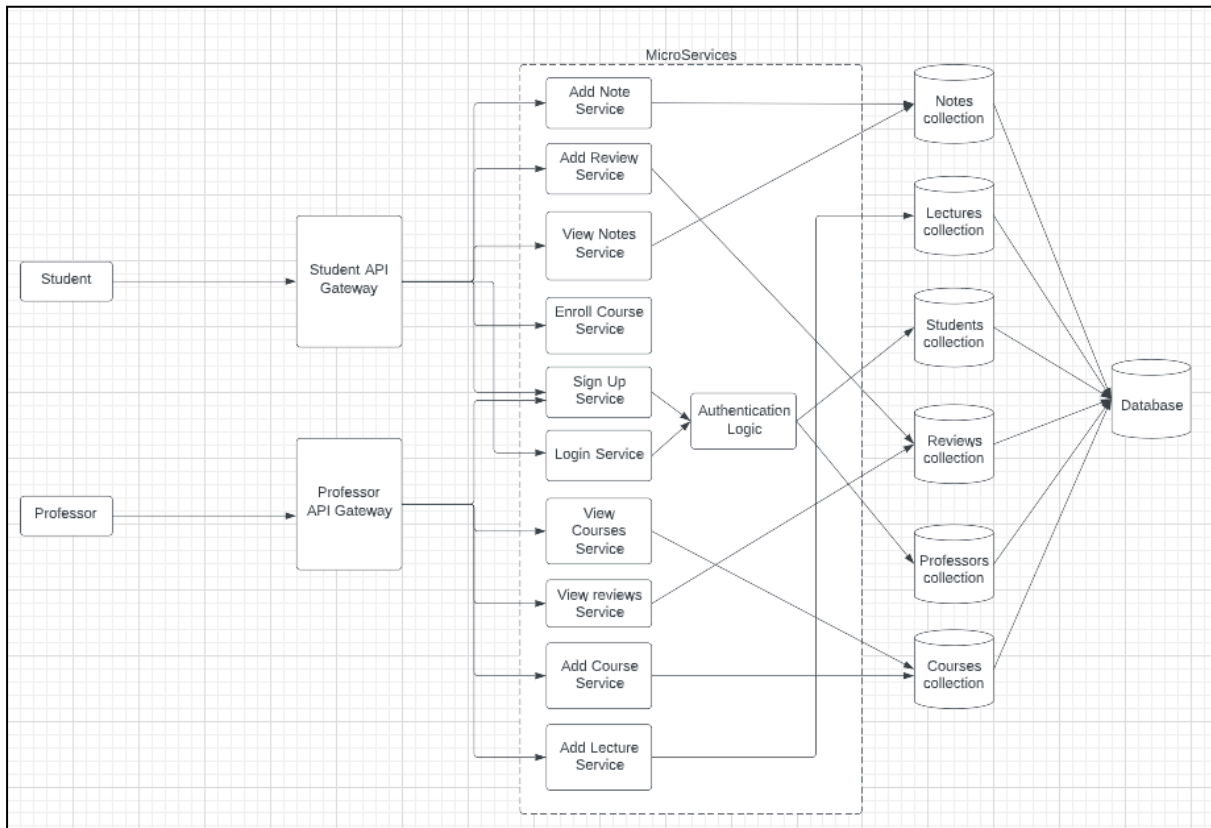
## **Referencing**

Referencing is used in the following classes:

- Course (profId)
- Review (courseId , lecNo)
- Lecture (courseId)
- Notes (lecNo , courseId, studentId)



# System Architecture



## Justification:

The proposed architecture is independently deployable and therefore allows us for continuous improvement and regular updates on the application. Also because there is minimal coupling in this architecture, a failure in a particular service will not cascade that failure to other services, thereby reducing the downtime. The independence among services lets us scale up or scale down specific services based on the requirements/modifications of some elements. The architecture also provides ease in resolving causes of performance issues and aids in testing.

# Conclusion

## Summary of the problem and work done

The aim of this project is to create a fully functional progressive web app that allows students to view lectures, take notes, and provide feedback that can be considered by the professors. We made a simple minimalistic user interface for the user to interact with easily. The user interface is almost similar for both students and professors. There are a few differences that are shown above in the user interface flows. The students can enroll in the required courses, view lectures and make notes and give reviews at particular timestamps of the lecture video. They can also control whether their notes can be seen by anyone or only by themselves. The professor can view the enrolled student details and feedback from the students at specific timestamps for a lecture.

## Summary of learning/reflections on project work

Obviously to start off with we learned a lot about the MERN stack in general. Our backend uses Express and NodeJs, frontend is built using React, and MongoDB for our database. We learned a lot about CSS styling techniques as well to make our website look better. The Material UI library from React was used extensively in order to improve the user interface. A lot of techniques such as making user stories, UI designing, Database designing, use case diagrams, and designing system architecture helped us greatly in creating a well-planned out website. We used LucidChart for User Interface and Database designing. We are using git as our version control system and GitHub as our repository. We have used Postman to test our api endpoints.

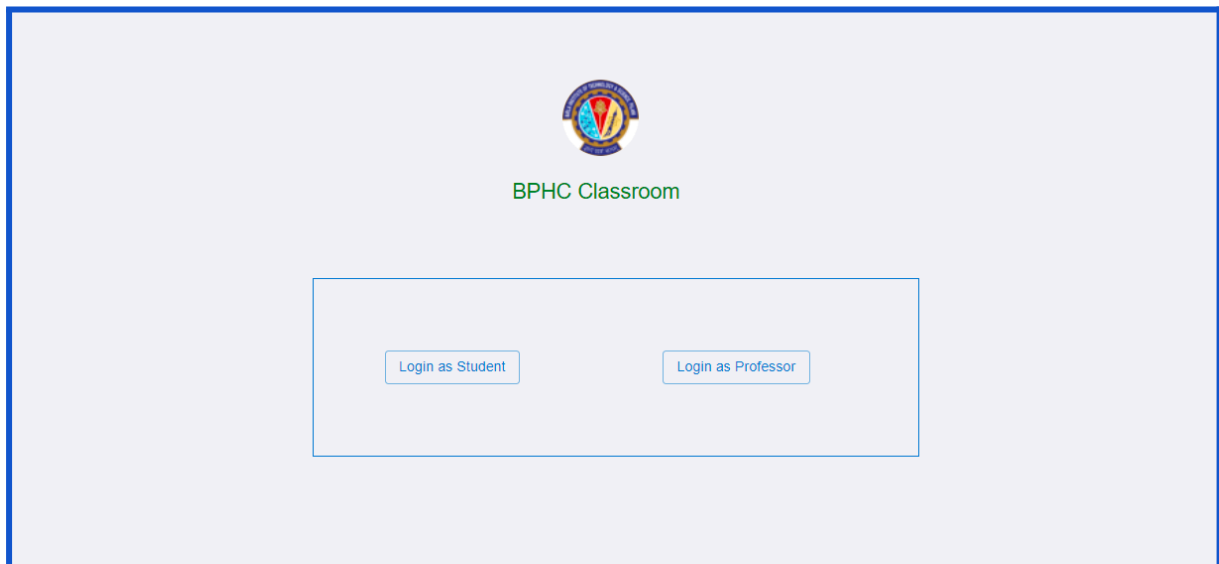
# Appendices


## A. Code Quality Report



## B. Sample Screenshots


Login and Signup pages:





Log In

[Don't have an account? Sign Up](#)



Sign Up


[Already have an account? Log In](#)

## Student dashboard (enrolled courses):

🏠 BPHC Classroom
⚙️

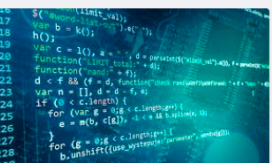
Hi Nitin, Your enrolled course(s) are

Enroll to New Course



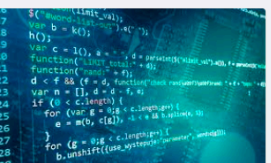
IS F341  
SE

Go to Course Content



CS F111  
Computer Programming

Go to Course Content



CS F320  
Data Science

Go to Course Content

## Selected Course Content:

BPHC Classroom				
CS F111 - Computer Programming				
Lec No	Date of Lecture	Title	Recording Link	Lecture Slides
1	01-01-2022	Introduction	<a href="#">Lecture Link</a>	<a href="#">Slides Link</a>
2	03-01-2022	Arrays	<a href="#">Lecture Link</a>	<a href="#">Slides Link</a>
3	05-01-2022	Pointers	<a href="#">Lecture Link</a>	<a href="#">Slides Link</a>

## Student's Lecture viewing page (notes):

BPHC Classroom

0.Agile

Back to Course Content

What is Agile Methodology?

THEY FOCUS ON:

- THINK
- INTERACT

TO ACHIEVE AGILITY

Watch on

Notes

Feedback

Add Notes at 0.00

ADD NOTES

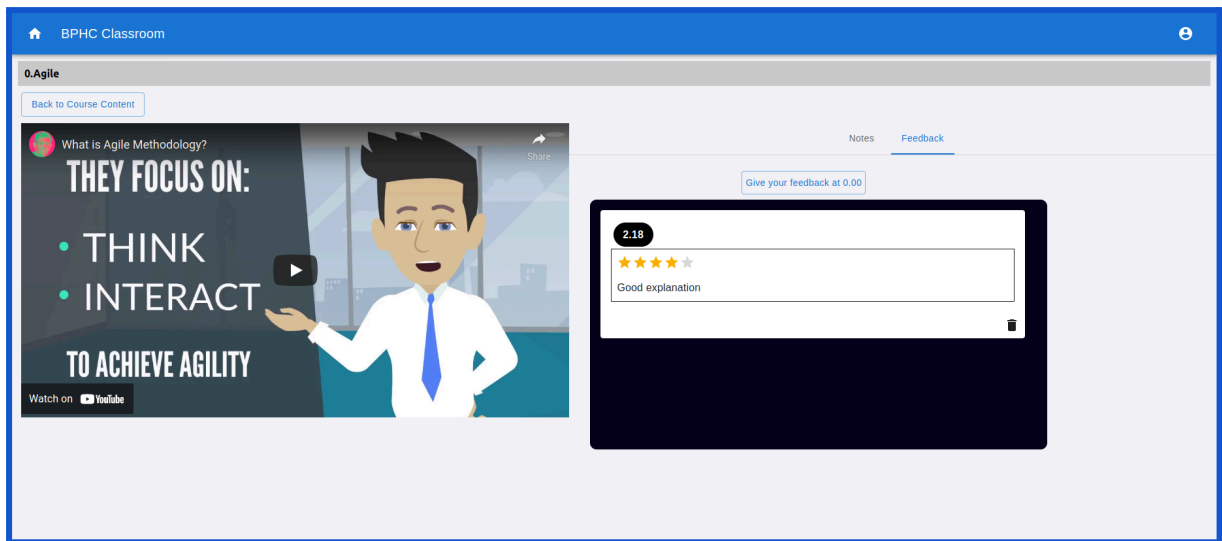
1.50

Sample note 2 from nitin public

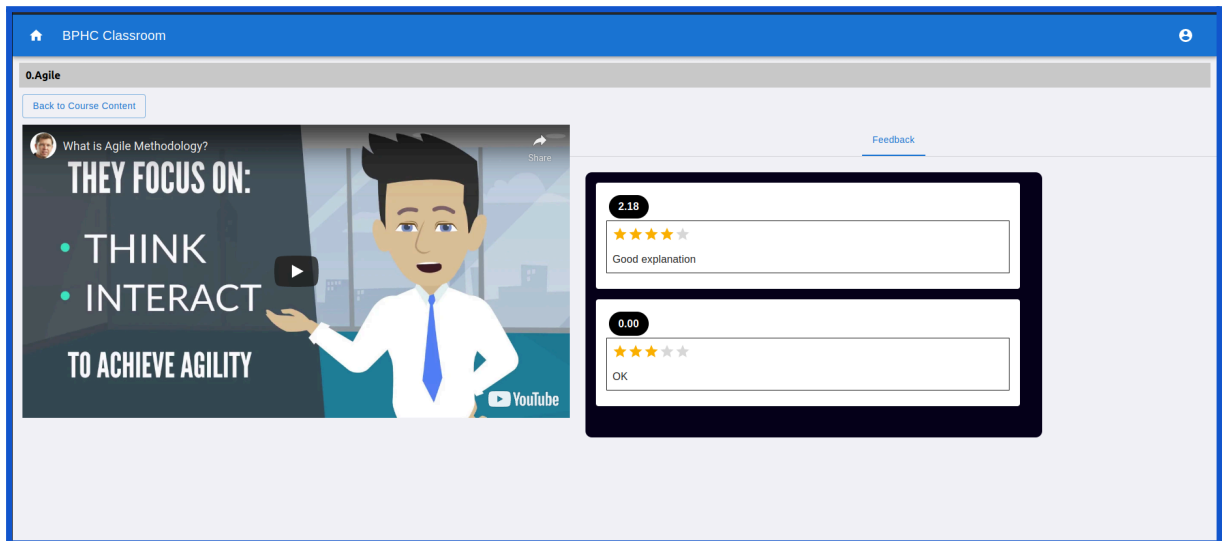
0.00

Sample note 1 from nivesh public

## Student's Lecture viewing (feedback):



## Professor's Lecture viewing page (feedback from all students):



Demo link : [link to recording](#)

