**College Notes Sharing Platform**

**Project submitted to the**

**APSSDC**

**Bachelor of Technology**

**In**

**Electronics and Communication Engineering**

**Nadimpalli Satyanarayana Raju Institute of Technology**

**Submitted By**

**Siriki ManojKumar – 22NU1A04A7**



**Under the guidance of**

**K.Narmada Mani**

**V. Vanitha**

**JUNE 2025**

**ABSRTACT**

The *College Notes Sharing Platform* is a cloud-integrated web application designed to streamline academic collaboration by enabling students to upload and access study materials with ease. Its primary aim is to bridge resource gaps and empower users with centralized, secure access to class notes. Focused on performance and user accessibility, the platform is built with responsiveness and scalability in mind. It serves as a digital bridge between contributors and learners in the academic ecosystem.

The frontend was developed using HTML and CSS, ensuring a clean and user-friendly interface across devices. Responsive design principles were applied to guarantee seamless browsing on both desktop and mobile. Emphasis was placed on clarity and usability, encouraging consistent engagement among users. Each component was styled for intuitive navigation and minimal cognitive load.

For backend data management, AWS DynamoDB was used to store and retrieve structured note metadata like titles, subjects, and contributors. The NoSQL nature of DynamoDB allows high-speed access and flexibility in managing dynamic content. Integration with AWS services ensures data durability, availability, and efficient scalability. It also supports seamless addition of new features like tagging and versioning in the future.

Static website hosting was enabled through Amazon S3, providing reliable, low-latency content delivery. Notes are uploaded and fetched through the S3 buckets, ensuring secure file storage and public access where necessary. The platform leverages S3’s static hosting to serve HTML pages directly, improving load times and reducing costs. This architecture ensures high availability with minimal operational overhead.

The solution is modular and future-ready, paving the way for enhancements like user authentication, rating systems, or advanced search filters. By leveraging modern web technologies and cloud-native tools, the platform offers an accessible, scalable, and maintainable solution for academic communities. It has the potential to evolve into a central knowledge hub. Overall, the project embodies a practical blend of usability and technical efficiency for academic collaboration.

**INTRODUCTION**

The *College Notes Sharing Platform* is a digital solution aimed at fostering collaborative learning in academic settings. With the growing need for accessible and organized study resources, this platform addresses a key student challenge. It offers a seamless method for sharing and retrieving lecture notes, assignments, and study guides. Students can contribute their content and benefit from a shared knowledge ecosystem.

The platform was conceptualized to support learners struggling with scattered or missing notes. It ensures that educational materials are preserved, easily accessed, and regularly updated. By allowing students to upload content, the system promotes peer-to-peer assistance. This approach encourages academic inclusion and reduces information disparity.

Built using HTML and CSS, the interface is designed for simplicity, speed, and clarity. Responsive design ensures compatibility with devices of different screen sizes. Users can quickly navigate subjects, view files, and locate relevant content. Visual hierarchy and styling emphasize accessibility and ease of interaction.Cloud-native tools such as AWS S3 and DynamoDB enable secure file hosting and fast retrieval. S3 ensures static web content is reliably served, while DynamoDB handles structured data efficiently. This architecture enhances scalability, allowing future features like tagging or smart search. Minimal backend maintenance means the focus stays on content sharing and usability.

Altogether, this platform provides a scalable, modern solution tailored to college communities. It combines intuitive design with robust cloud infrastructure for a powerful user experience. The project reflects a commitment to educational equity through technological innovation. With future upgrades in mind, it sets the foundation for a wider academic impact.

**SYSTEM REQUIREMENTS:**

**SOFTWARE REQUIREMENTS**

* **HTML** – For building the structure of the user interface.
* **CSS** – For styling and creating a responsive front end.
* **AWS Dynamo DB --**It’s designed for high performance at scale, offering

low-latency data access.

**HARDWARE REQUIREMNTS:**

* **Processor:** Dual Core 2.0 GHz or higher.
* **RAM:** Minimum 2 GB (4 GB recommended for hosting environment).
* **Storage:** 20 GB free disk space (additional space as per number of uploaded files).
* **Network:** Stable broadband internet connection for cloud integration.

**COMPONENTS OF THE PROJECT**

**HTML:**

In this project, I used HTML to structure every part of the web application, including the homepage, upload forms, and notes display sections. It provided the basic skeleton that defined how content appears on each page. I used semantic tags like <header>, <section>, and <article> for better accessibility and clarity. Form elements handled user inputs for uploading files and signing in. Tables organized notes by subject and semester, while links allowed easy navigation across resources. HTML5 features enhanced interactivity without external scripts. I also included placeholders and labels to improve the user experience. Overall, HTML formed the structural core of the platform.

**CSS :**

CSS was applied to bring visual appeal and user-friendly design to the platform. I customized fonts, colors, margins, and paddings to align with a clean academic style. Flexbox and Grid helped create a responsive layout that worked well across devices. Transitions and hover effects made buttons and links more interactive. Media queries were used to ensure proper rendering on mobile, tablet, and desktop screens. External stylesheets kept the CSS modular and easy to maintain. Animations provided feedback during file uploads or when errors occurred. With CSS, the overall user interface became more polished and accessible.

**Amazon S3** :

Amazon S3 was used as the main cloud storage service for hosting uploaded notes. I created and configured S3 buckets to categorize notes by course and semester. Files were uploaded using secure API endpoints and retrieved through generated URLs. I enabled versioning to track changes and allow file updates. Bucket permissions were set to restrict unauthorized access while supporting temporary downloads. S3’s reliability ensured consistent data availability and durability. It integrated well with the backend logic to automate file management tasks. This made S3 a scalable and efficient storage solution for the platform.

**AWS Dynamo DB:**

AWS DynamoDB is a fully managed NoSQL database service by Amazon.It’s designed for high performance at scale, offering low-latency data access.It stores data as key-value pairs or documents.Great for apps requiring millisecondresponse times at any scale.It automatically scales throughput capacity and storage.Supports event-driven programming via DynamoDB Streams.

**ARCHITECTURE:**

The platform follows a modular, cloud-based architecture combining frontend, backend, and storage layers. HTML and CSS power the user interface, while Amazon S3 handles file storage securely in the cloud. Ubuntu serves as the backend OS, managing application logic and deployment processes. This layered design ensures scalability, reliability, and a seamless user experience.

### **1. Project HTML Code**

The HTML code forms the structural backbone of the College Notes Sharing Platform. It defines page elements like headings, file upload fields, and subject filters for users. Each section is organized to promote clarity and ease of navigation. Forms are created for uploading notes, along with dynamic content sections to display data. HTML tags are semantically chosen to enhance accessibility and search performance. Combined with CSS, the layout becomes visually consistent and user-friendly.

### **2. Tables Creation in Amazon DynamoDB**

Amazon DynamoDB is used to create NoSQL tables that store metadata for each note. Typical tables include attributes like note\_id, subject, contributor, filename, and upload\_date. These tables allow fast read/write operations with minimal latency, ideal for dynamic platforms. The schema-less nature of DynamoDB supports evolving data structures as features are added. Partition keys and sort keys are carefully selected to support efficient querying and filtering. The tables work seamlessly with AWS SDK to integrate with the frontend.

### **3. Notes Metadata**

Notes metadata refers to the descriptive details that accompany each uploaded file. This includes subject name, semester, uploader’s ID, and a brief title or description. Metadata helps categorize content, making it easier for users to search or filter notes. It’s stored in DynamoDB, ensuring fast, indexed access for display or updates. This information is critical for organizing and retrieving files contextually. Proper metadata boosts user experience and supports future features like rating or tagging.

### **4. General Purpose Bucket**

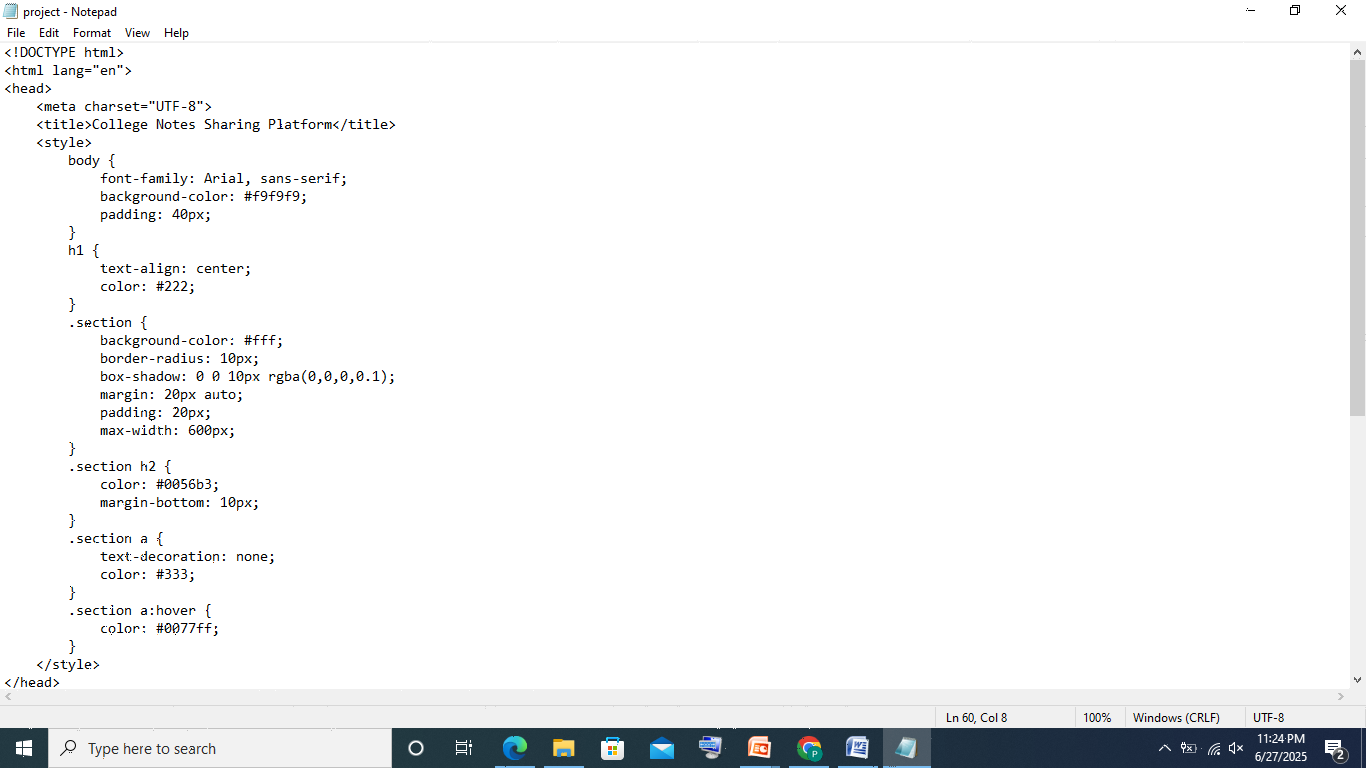
A general-purpose Amazon S3 bucket stores all note files uploaded by users. Each file is assigned a unique key and stored securely with appropriate permissions. The bucket supports static hosting, letting users view HTML pages via S3 URLs. Files are accessible through presigned URLs or public links based on access rules. Versioning and lifecycle policies can be enabled to manage file aging and storage. S3 ensures reliable delivery and acts as the content delivery backbone of the project.

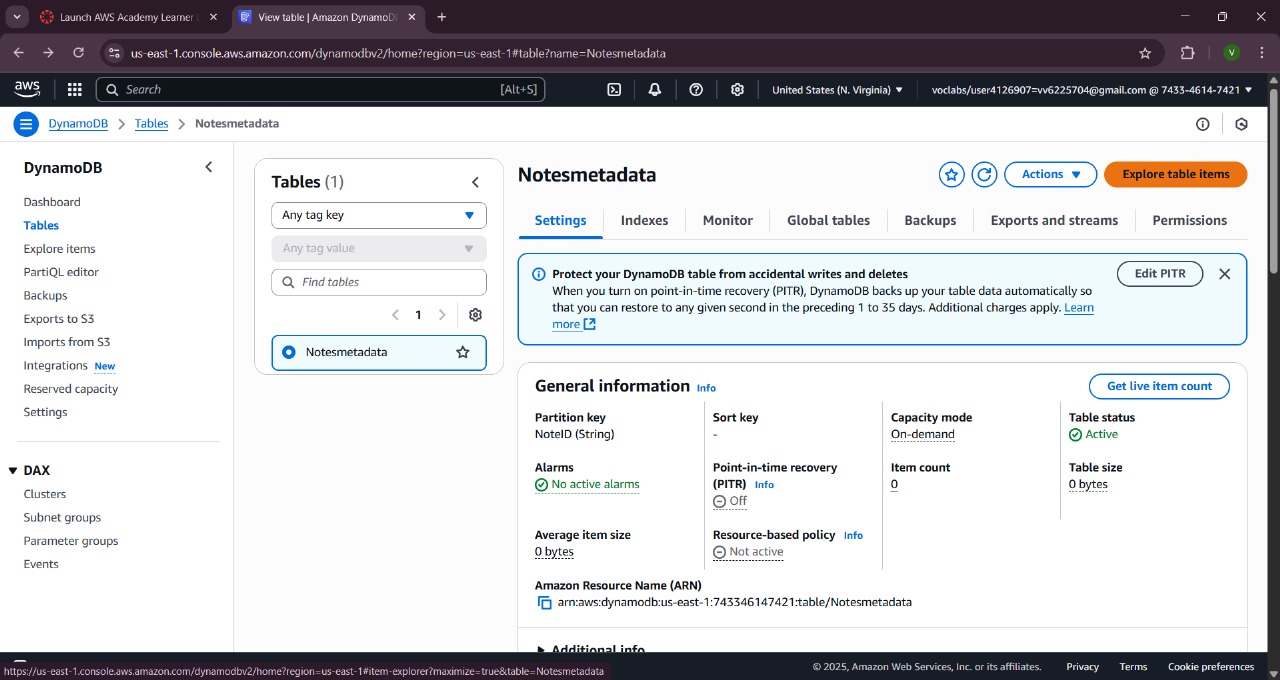
### **5. College Notes Sharing Platform**

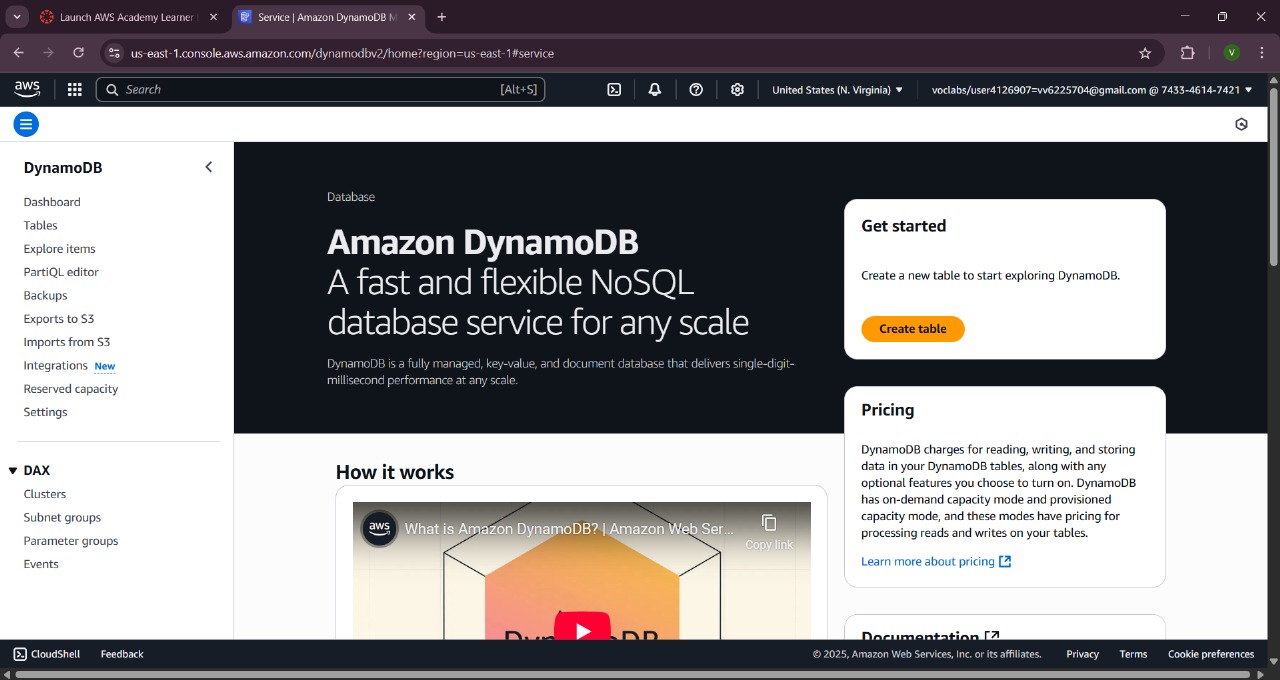
This platform is a student-centric tool built to foster academic content sharing. It combines frontend technologies like HTML/CSS with AWS cloud resources. Students can upload, browse, and download notes from a centralized interface. Cloud services provide a balance of security, scalability, and performance. Its modular design supports new features like advanced search or login systems. Overall, it serves as a digital hub for collaborative and accessible learning.

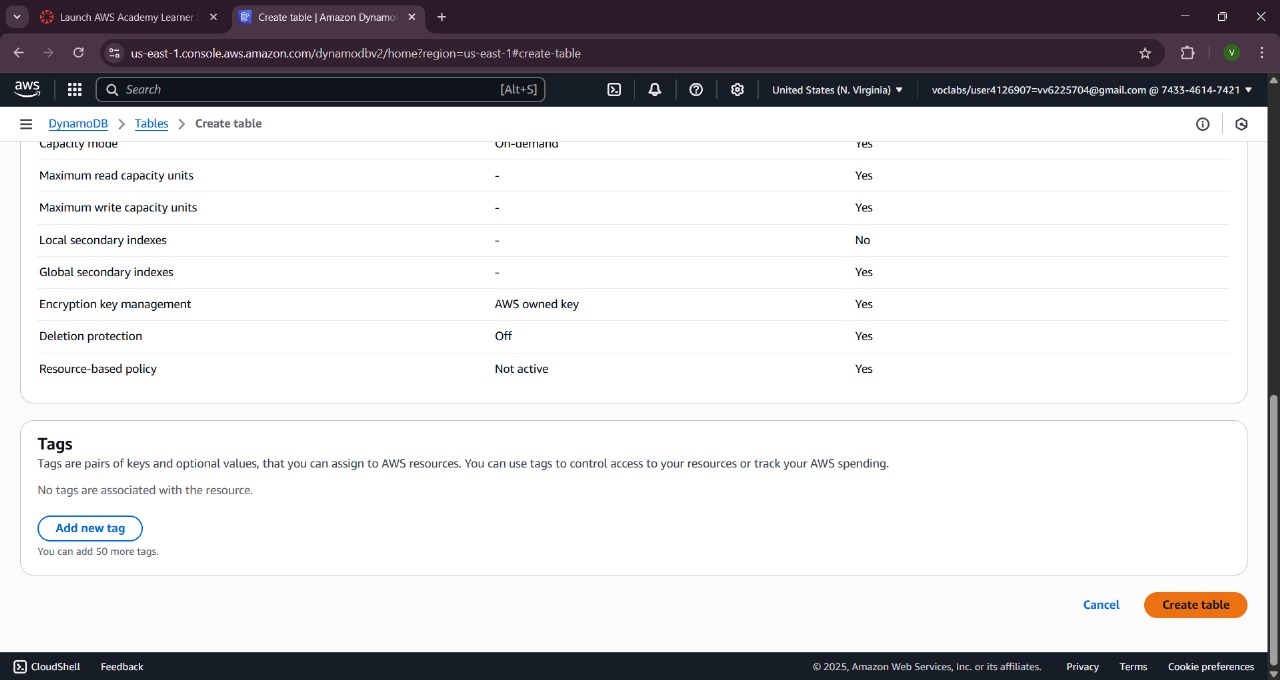
.

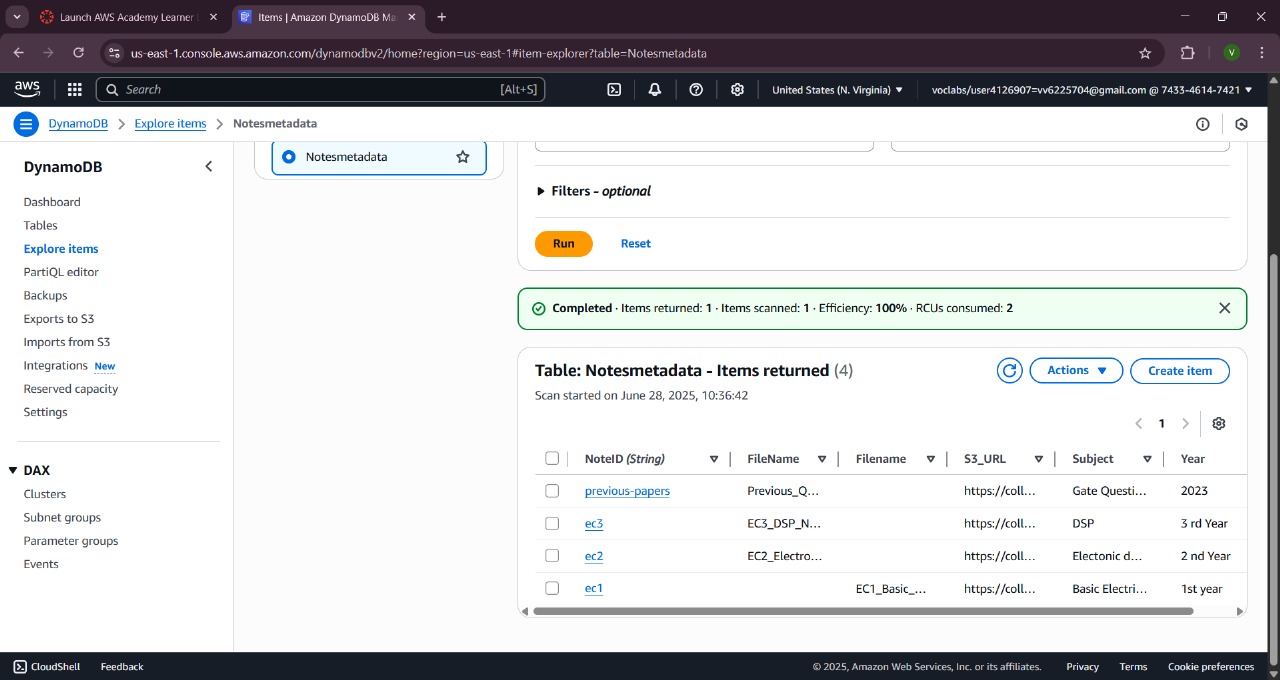
**PROJECT CODE**

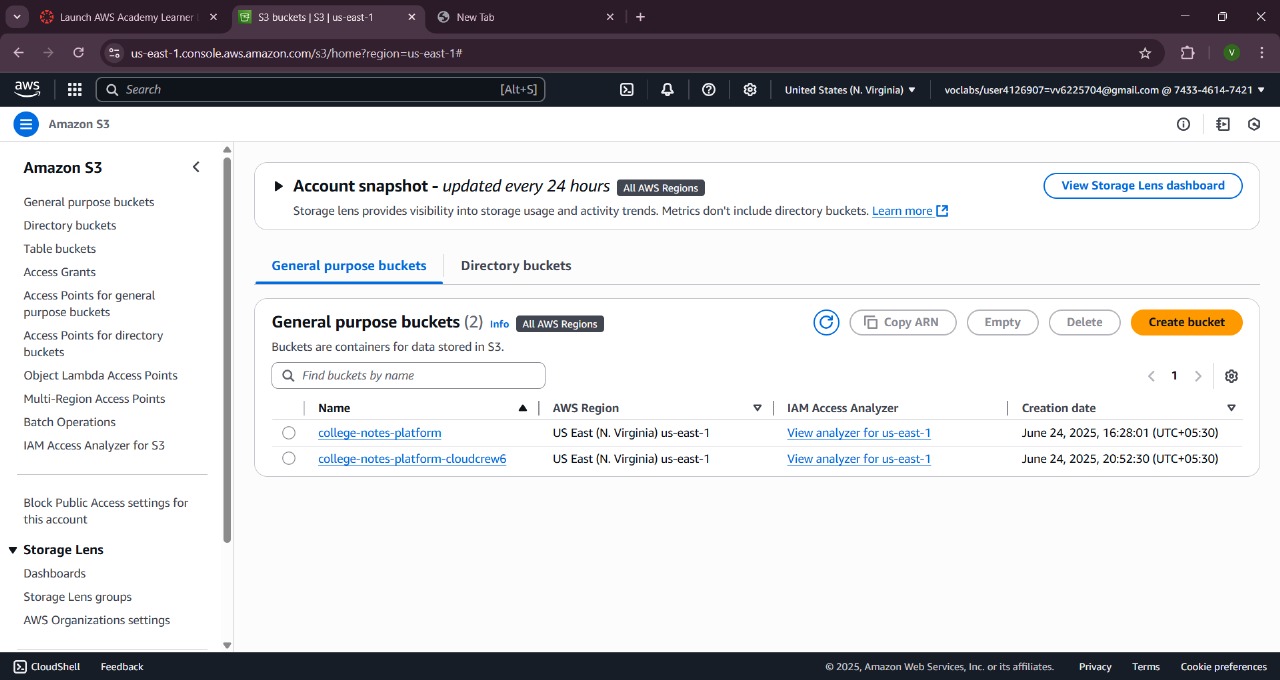


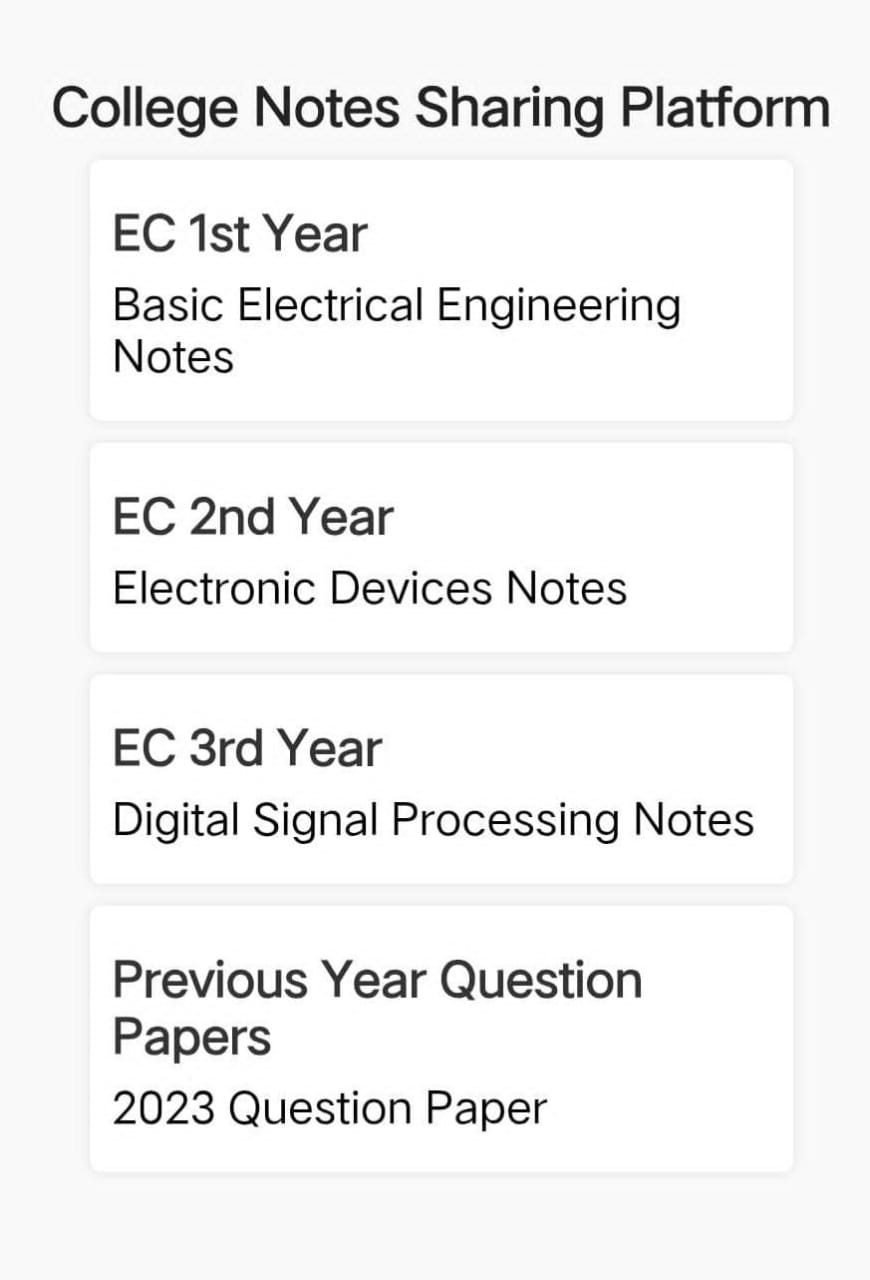




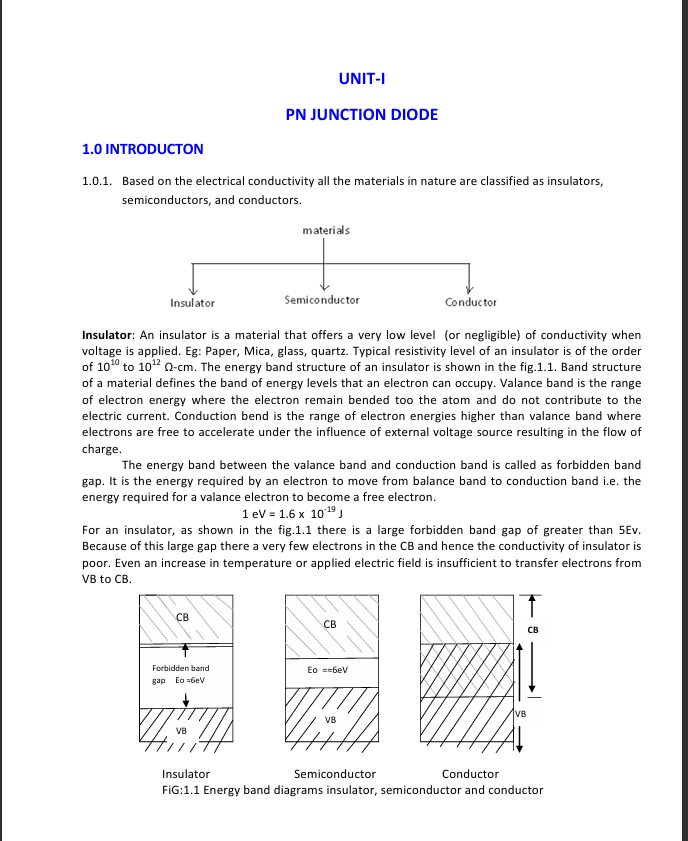






****





**ADVANTAGES**

**1.Enhanced Academic Collaboration** Promotes peer-to-peer learning by allowing easy sharing of study materials. Fosters an inclusive learning environment where everyone benefits.

**2. Scalable Cloud Infrastructure** AWS S3 and DynamoDB provide a robust, scalable backend. Supports growing user data without compromising performance.

**3. User-Friendly and Responsive Interface** HTML and CSS ensure a clean, intuitive design across devices. Users can browse and access content with minimal effort.

**4. Secure and Reliable File Hosting** Amazon S3 offers fast, encrypted, and dependable storage. Ensures smooth and secure access to uploaded notes.

**5. Easy Deployment and Low Maintenance** Static hosting reduces backend complexity and server load. Speeds up deployment and simplifies platform management.

**6. Potential for Expansion and Customization** Supports future features like filtering, tagging, or login systems. Keeps the platform flexible, modern, and user-focused

**BENEFITS**

**1. Centralized Knowledge Repository** All academic notes are stored in one place, minimizing fragmentation. Students save time searching for reliable content. It builds a dependable source for exam preparation.

**2. Easy Integration with Cloud Services** AWS S3 and DynamoDB make the backend seamless and scalable. They ensure low-latency operations and durable file storage. This allows smooth performance even with growing usage.

**3. Responsive and Accessible Design** HTML and CSS ensure the platform works well across devices. It maintains layout consistency for phones, tablets, and desktops. Students can access notes anytime, anywhere.

**4. Promotes Collaborative Learning** Students contribute and benefit from shared resources equally. It fosters a sense of academic community and teamwork. Collaboration enhances understanding through multiple perspectives.

**5. Reliable Data Management** DynamoDB stores structured metadata for each uploaded note. This supports fast searches and efficient categorization. Future enhancements like filters and tagging are easy to implement.

**6. Secure and Controlled File Access** AWS S3 secures files with encryption and permission rules. This safeguards user-uploaded content from unauthorized access. It builds user trust and ensures compliance.

**7. Simplified Upload and Retrieval Process** Uploading notes takes just a few clicks through a clear interface. Files can be browsed and downloaded quickly and easily. This encourages consistent participation among students.

**8. Encourages Technological Awareness** Students interact with real-world cloud tools and workflows. This fosters technical literacy and curiosity in cloud computing. It bridges academics and practical tech exposure.

**9. Low Maintenance and Cost-Effective** Static hosting reduces hosting fees and server-side complexity. The system is easy to update and extend without heavy workloads. This makes it sustainable for long-term use.

**10. Room for Continuous Improvement** The modular design supports new features like logins or dashboards. It can evolve with student needs and tech trends. This future-proofing ensures long-term relevance.

**CONCLUSION**

The *College Notes Sharing Platform* stands as a practical solution to the persistent challenge of academic resource sharing. By combining a simple user interface with powerful cloud technologies, it bridges the gap between availability and accessibility. The use of HTML and CSS ensures that users can interact easily, regardless of their device or technical ability. Meanwhile, AWS S3 and DynamoDB provide a strong foundation for data reliability and security. Together, these components create a seamless user experience with minimal maintenance overhead. The result is a high-impact platform tailored to real educational needs.

Beyond technical efficiency, the project fosters a spirit of collaboration and collective growth among students. It encourages users to contribute their knowledge, reducing information silos within academic environments. By enabling wide participation, the platform promotes educational inclusivity and equal opportunity. Its design supports scalability, allowing for future enhancements without structural rework. Features like tagging, search filters, and authentication can be smoothly integrated going forward. This adaptability ensures the platform can evolve alongside user needs and feedback.

In summary, the project reflects both technical strength and a thoughtful understanding of academic challenges. It delivers a cost-effective, scalable, and user-centric solution through cloud-native tools and intuitive design. The platform not only facilitates efficient note sharing but also cultivates a digital community for learners. Its architecture balances function with future readiness, setting the stage for further innovation. The project demonstrates how technology can transform learning experiences when applied mindfully. Ultimately, it’s a step toward democratizing education through accessible digital platforms.

**REFERENCE WEBSITES**

1. **HTML & CSS**

* [MDN Web Docs](file:///C:\Users\Manu\AppData\Local\Microsoft\Windows\INetCache\IE\QLPPSVNF\HTML%20&%20CSS) – HTML: Comprehensive guide to HTML elements and usage.
* [MDN Web Docs](file:///C:\Users\Manu\AppData\Local\Microsoft\Windows\INetCache\IE\QLPPSVNF\HTML%20&%20CSS) – CSS: In-depth documentation on CSS styling, layout, and responsiveness.

**2. Amazon S3**

* [Hosting a Static Website on Amazon S3](file:///C:\Users\Manu\AppData\Local\Microsoft\Windows\INetCache\IE\QLPPSVNF\Hosting%20a%20Static%20Website%20on%20Amazon%20S3) – AWS Docs: Official guide to configuring S3 for static website hosting.
* [Deploy Static Websites to Amazon S3](file:///C:\Users\Manu\AppData\Local\Microsoft\Windows\INetCache\IE\QLPPSVNF\Deploy%20Static%20Websites%20to%20Amazon%20S3) – DEV Community: Step-by-step tutorial with practical deployment tips.
* [GitHub Project: S3 Static Website Example](file:///C:\Users\Manu\AppData\Local\Microsoft\Windows\INetCache\IE\QLPPSVNF\GitHub%20Project:%20S3%20Static%20Website%20Example): Sample project using HTML, CSS, and S3 for hosting.

**3. Ubuntu**

* [Ubuntu Server Guide](file:///C:\Users\Manu\AppData\Local\Microsoft\Windows\INetCache\IE\QLPPSVNF\Ubuntu%20Server%20Guide): Official documentation for deploying and managing Ubuntu servers.
* [DigitalOcean](file:///C:\Users\Manu\AppData\Local\Microsoft\Windows\INetCache\IE\QLPPSVNF\DigitalOcean)– How to Deploy a Flask App on Ubuntu: Great for backend integration if you're using Python.