IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Reportease: A Modern Report Generation Tool

[1]dr. Sunita Chalageri, [2]abhiram Ys, [3]keerthika S, [4]gaana S, [5]dhruthi Umesh S

[1]associate Professor, [2]student, [3]student, [4]student, [5]student

[1]dept Of Cse,

[1]k.S. Institute Of Technology, Affiliated To Vtu, Bengaluru, Karnataka

Abstract: This Solution Introduces A Web-Based Report Generation Tool That Streamlines Academic Writing By Integrating Advanced Formatting And Data Management Features. The User- Friendly Interface Supports Autoformatting, Including Standardized Font Sizes, Justified Content, And 1.5-Line Spacing, While Offering Seamless Data Input And Session Management With Auto-Save Functionality. Aws Integration Ensures Secure Storage And Processing, And The Tool Includes Robust Data Management Capabilities Such As Historical Access To Previously Generated Documents. This Comprehensive Solution Aims To Enhance Student Productivity And Reduce Stress Associated With Academic Report Creation By Providing An Efficient, Reliable, And Userfriendly Platform That Adheres To Academic Standards.

Keywords: Report Generation Tool, Web-Based Application, Aws Integration, Academic Support, Data Management, User-Friendly Interface, Document, Google Ai Studio Apis

I. Introduction

In Academic And Research Environments, The Process Of Report Writing Plays A Crucial Role In Documenting Findings, Analyses, And Insights. However, Students And Researchers Often Face Significant Challenges When Creating Structured And Well-Formatted Reports That Comply With Institutional Guidelines. Traditional Word Processing Tools Such As Microsoft Word And Google Docs Are Widely Used But Require Substantial Manual Effort To Ensure Proper Formatting. On The Other Hand, Latex, A Powerful Typesetting System, Offers Advanced Formatting Capabilities But Has A Steep Learning Curve, Making It Inaccessible To Many Students Unfamiliar With Markup-Based Document Generation.

Recognizing These Challenges, We Introduce Reportease, A Web-Based Report Generation Tool Designed To Streamline And Enhance Academic Writing. Reportease Offers A User-Friendly Interface That Automates Formatting, Ensuring Compliance With Academic Standards Such As Standardized Font Sizes, Justified Content, And Consistent Line Spacing. The Platform Provides A Structured Workflow, Allowing Users To Easily Input And Organize Content Into Different Sections And Chapters. Additionally, Reportease Integrates Auto-Save Functionality And Session Management, Preventing Data Loss And Improving User Convenience.

To Further Enhance Usability And Efficiency, Reportease Incorporates Aws Cloud Services For Secure Data Storage And Scalable Processing. The Platform Allows Students To Access Previously Generated Documents, Reducing Redundancy And Improving Workflow Efficiency. A Part Of Innovation In Reportease Is Jilang, A Custom Markup Language That Simplifies The Process Of Structured Report Generation. Jilang Combines The Ease Of Use Of Jsonlike Syntax With Powerful Formatting Capabilities, Enabling Users To Focus On Content Creation Rather Than Manual Formatting Adjustments.

This Paper Presents The Architecture, Design, And Implementation Of Reportease, Highlighting Its Advantages Over Traditional Document Creation Tools. We Explore Its Frontend And Backend Development, Aws Integration, And The Role Of Jilang In Optimizing Academic Report Generation. Furthermore, We Discuss The Ai-Powered Report Generation Feature, Which Utilizes Gemini Api To Create Structured, High-Quality Reports With Minimal User Intervention. The Proposed Solution Aims To Reduce The Time And

Effort Required For Report Writing While Ensuring Compliance With Academic Standards, Thereby Enhancing Student Productivity And Reducing Stress Associated With Academic Documentation. Through This Study, We Aim To Demonstrate How Reportease Serves As An Innovative And Efficient Solution For Modern Academic Report Writing. The Paper Concludes With An Evaluation Of Its Performance, Key Findings, And Potential Enhancements For Future Development, Such As Integrating Grammar And Plagiarism Checking, Multi-Format Conversion, And Collaborative Editing Capabilities.

Ii.Literature Review

"Graphql: A Systematic Mapping Study" This Study [1] Examines The Adoption, Trends, And Challenges Associated With Graphql, Emphasizing Its Role In Optimizing Data Retrieval. Drawing From These Insights, Reportease Incorporates Graphql Alongside Apollo Client To Enhance Data Management. This Integration Ensures That Only The Necessary Data Is Retrieved, Reducing Redundancy And Improving The Efficiency Of The Report Generation Process. By Streamlining Data Handling, Reportease Enhances Performance And Responsiveness, Making Report Creation More Seamless And Effective.

"Test Report Generation Using Json" This Study [2] Explores The Benefits Of Using A Lightweight And Human-Readable Data Format For Organizing Report Content And Automating The Generation Process. It Emphasizes How Such Formats Enhance Readability, Ease Of Use, And Processing Efficiency. Reportease Incorporates These Principles Through Jilang, A Custom-Designed Language With A Json-Like Syntax, Allowing For Structured Data Representation While Maintaining Simplicity. By Leveraging Jilang, Reportease Enhances User Experience, Streamlines Formatting, And Ensures Efficient Handling Of Report Data, Ultimately Making The Report Creation Process More Intuitive And Accessible.

"Research Articles In Simplified Html: A Web-First Format For Html-Based Scholarly Articles" This Study [3] Explores The Benefits Of Using A Simplified Markup Language To Improve Content Creation And Structured Document Formatting. It Emphasizes How Such Languages Enhance Readability, Organization, And Ease Of Use While Reducing The Complexity Of Manual Formatting. Reportease Adopts A Similar Approach With Jilang, A Custom-Designed Language Featuring A Json-Like Syntax, To Streamline Report Structuring. By Integrating Jilang, Reportease Improves User Accessibility, Automates Formatting Tasks, And Ensures Efficient Data Processing, Ultimately Making Report Generation More Seamless And Effective.

"Gemini-The Most Powerful Llm: Myth Or Truth" This Study [4] Explores The Use Of Advanced Language Models For Improving Tasks Like Text Processing And Generation. Reportease Applies Similar Principles By Incorporating Automation To Assist In Content Creation, Ensuring Reports Are Well-Structured, Coherent, And Easier To Generate, Ultimately Simplifying The Writing Process For Students.

"Analyzing Next.Js's Effect On Search Engine Optimization And Website Speed (Seo)" This Study [5] Examines How Next.Js Improves Website Performance And Search Engine Optimization Through Server-Side Rendering, Static Site Generation, And Automatic Code Splitting. Reportease Incorporates These Features To Enhance Loading Speed, Ensure Seamless Navigation, And Improve Accessibility. By Leveraging Next.Js, The Platform Provides A More Efficient And User-Friendly Environment For Report Creation.

"Enhancing Seo In Single-Page Web Applications In Contrast With Multi-Page Applications" This Study [6] Discusses The Impact Of Next.Js On Website Performance And Search Engine Optimization, Focusing On Features Such As Server-Side Rendering, Static Site Generation, And Automatic Code Splitting. Reportease Incorporates These Capabilities To Improve Loading Speed, Enhance User Experience, And Ensure Better Accessibility, Making The Report Creation Process More Efficient And Streamlined.

"Essential Elements Of Writing A Research/Review Paper For Conference/Journals" This Study [7] Provides Key Insights Into Structuring And Formatting Research Papers, Covering Aspects Such As Content Organization And The Publishing Process. While Not Directly Influencing Reportease's Functionality, It Offers Valuable Guidelines For Creating Well-Structured Reports. Applying These Principles Can Enhance Report Clarity And Readability, Ultimately Improving The User Experience Of The Report Generation Tool.

"Basics Of Research Paper Writing And Publishing" The Study [8] Highlights How Crucial Clear Structure And Clarity Are When It Comes To Academic Writing. Reportease Takes These Ideas A Step Further By Offering Ready-To-Use Templates—Such As The Ieee Format—So Users Can Easily Create Well-Structured, Publication-Quality Reports Without Having To Start From Scratch.

"Using Json For Data Exchanging In Web Service Applications" This Study [9] Highlights The Advantages Of Using Json For Data Exchange In Web Services, Mainly Due To Its Lightweight Nature And Readability. Building On This Idea, Reportease Uses Json To Efficiently Transfer Structured User Input Between The Frontend And Backend, Allowing For Smooth Interaction And Faster Report Generation.

Iii. Software Requirements

- Aws Client-S3: Used For Secure And Scalable Cloud Storage To Manage Report-Related Data And Files. Reportease Leverages Aws S3 To Store Generated Reports, Ensuring Accessibility And Backup While Maintaining Data Security.

- Google-Generative-Ai: Enables Ai-Powered Assistance Within Reportease By Integrating Google's Ai Tools. This Allows For Features Such As Automated Content Generation, Intelligent Suggestions, And Context-Based Enhancements, Improving The Overall Report-Writing Experience.

- Apollo-Server-Express: Acts As A Bridge Between Graphql And Express.Js, Allowing Reportease To Efficiently Handle Data Queries And Mutations. This Enables Users To Fetch Only The Required Report Data, Reducing Unnecessary Processing And Improving System Performance.

- Express: Serves As The Backend Framework, Managing Api Requests, Handling User Authentication, And Ensuring Smooth Communication Between The Frontend, Database, And Cloud Services. Reportease Relies On Express.Js For Routing, Session Management, And Api Endpoint Creation.

- Graphql: Used To Define And Query Report-Related Data Efficiently. By Allowing Precise Data Retrieval, Graphql Ensures That Reportease Fetches Only Relevant Information, Improving Performance And Reducing Redundant Requests.

- Passport: Integrated For Secure User Authentication, Enabling Features Like Google Sign-In And Account Verification. Reportease Uses Passport.Js To Ensure Only Authorized Users Can Access Stored Reports And Manage Their Data Securely.

- Next.Js: Powers The Frontend Of Reportease, Combining The Best Features Of React With Server-Side Rendering For Improved Performance. This Ensures Fast Page Loads And A Seamless User Experience While Providing An Efficient Framework For Building Dynamic Reports.

- Tailwind Css: Simplifies The Styling Of Reportease's Ui, Enabling A Responsive And Visually Appealing Interface.

- Typescript: Helps Maintain Code Quality And Reliability By Reducing Errors Through Strict Type Checking. In Reportease, Typescript Enhances The Development Process By Making The Code More Structured, Readable, And Easier To Debug, Leading To A More Robust And Maintainable Application.

Iv. System Design

The System Architecture Of The Report Generation Tool Is Designed To Ensure Scalability, Security, And Ease Of Use. The Primary Components Of The System Include The Frontend, Backend, And Data Storage.

A. Frontend

The Frontend Is Developed Using Nextjs Web Framework. It Provides A User-Friendly Interface For Data Input, Report Formatting, And Reordering Of The Added Data. Key Features Of The Frontend Include Intuitive Ui For Easy Data Entry And Formatting, Reordering Added Data, Auto-Save Functionality And Session Management.

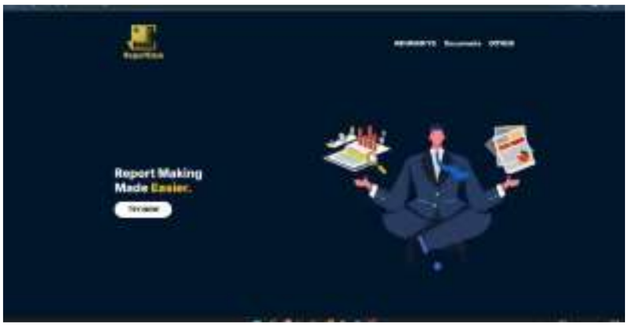


Fig. 1: Home Page

Built With Next.Js, The Frontend Provides A Responsive And Interactive User Interface. It Enables Users To Input, Edit, And Generate Reports Effortlessly. Tailwind Css Is Used For Styling To Ensure A Clean And Modern Look.

The Home Page Is The Initial Interface Users Encounter. Figure 1 Highlights The Main Features Of Reportease And Provides Easy Access To Start Generating Reports.

B. Backend

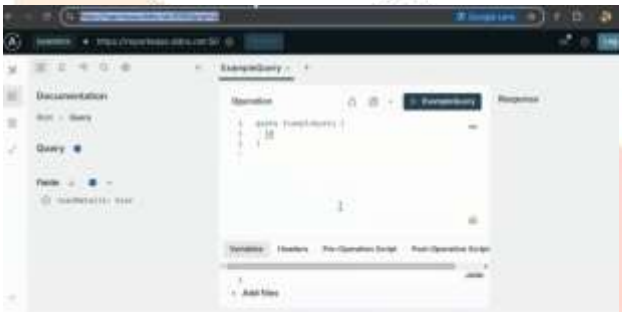
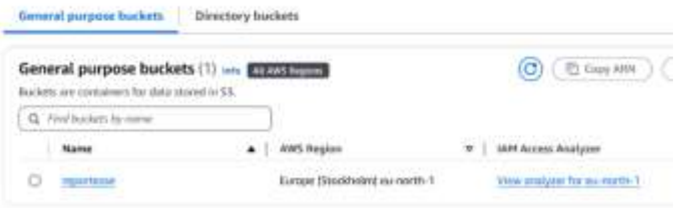


Fig. 2: Graphql Backend

The Backend Is Powered By Node.Js And Express.Js, Handling Api Requests, User Authentication, And Data Processing As Shown In Figure 2. It Integrates Graphql With Apollo Server, Allowing Efficient Data Querying And Reducing Redundant Network Requests.



General purpose buckets | Directory buckets

General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	IAM Access Analyzer
reportease	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1

Fig. 3: S3 Bucket

Database & Storage: Reportease Utilizes Aws S3 As Shows In Figure 3 For Secure Storage Of Reports And User Data, Ensuring Easy Retrieval And Version Control.

C. Core Functionalities & Features:

Auto-Save & Session Management: Prevents Data Loss By Storing Progress In Real-Time.

Ai-Assisted Report Generation: Integrates Google Ai Studio Apis To Generate Content Suggestions And Enhance Document Structure.

Jilang Markup Integration: A Custom Json-Like Syntax Simplifies Structured Report Generation, Reducing Manual Formatting Efforts.

Latex Editor Integration: Users Can Refine And Customize Their Final Output Using The Built-In Latex Editor, Allowing For Greater Control Over Research Paper Formatting.

Predefined Formatting Templates: To Ensure Compliance With Academic Guidelines, Reportease Automates Document Formatting With Ieee Standards And General Report Format.

V. Methodology

The Methodology Of Reportease Focuses On A Structured Approach To Automating And Optimizing Academic Report Generation. The Implementation Follows A Systematic Workflow, Integrating Various Technologies To Enhance Efficiency, Accuracy, And User Experience. The Methodology Is Broken Down Into The Following Key Phases:

1. Requirements Analysis

Before Development, A Comprehensive Analysis Was Conducted To Identify Common Challenges In Academic Report Writing. The Primary Requirements Identified Were:

- Standardized Formatting For Academic Compliance.
- Automated Content Structuring And Generation.
- Cloud Storage And Document Retrieval.
- Ai-Driven Content Assistance

2. System Development Approach

The Development Process Follows An Agile Methodology As Shown In Figure 4, Ensuring Iterative Improvements Based On User Feedback. The System Is Divided Into Frontend And Backend.

Frontend Development

- Implemented Using Next.Js And Tailwind Css For Dynamic And Responsive Ui.
- Provides A Structured Input System For Users To Organize Report Sections.
- Features Real-Time Preview And Drag-And-Drop Functionality For Easy Content Management.

Backend Development

- Built With Node.Js And Express.Js To Handle Api Requests And User Authentication.
- Uses Graphql With Apollo Server For Efficient Data Querying And Minimal Server Load.
- Manages Session Handling And Auto-Save Functionality To Prevent Data Loss.

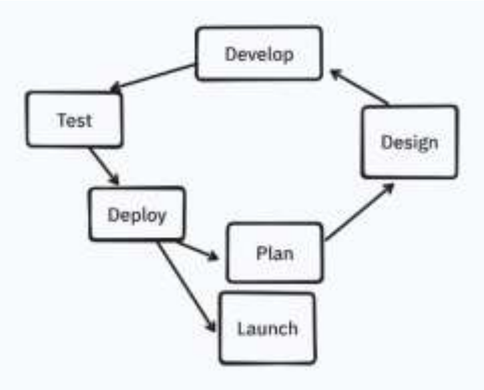


Fig. 4: Agile Development

3. Testing & Validation

The System Undergoes Rigorous Testing In Multiple Stages:

- Unit Testing: Each Component (Frontend, Backend, Ai Features) Is Tested Separately.
- Integration Testing: Ensures Seamless Interaction Between Frontend, Backend, And Ai Services.

The Development Of Reportease Follows A Wellorganized, Scalable, And Performance-Focused Methodology. It Is Built To Enable Authenticated Users To Create Polished, Professional Reports Using A Selection Of Pre-Designed Templates. These Templates Support Various Content Elements, Including Paragraphs, Bullet Etc.

Users Log In Using Google Authentication, Implemented Via Passport.Js. After Successful Authentication, Their Login Information Is Securely Saved In A Postgresql Database. Once Logged In, Users Are Directed To The Homepage, Where They Can Start Creating Their Reports.

There Two Different Methods For Generating Latex-Based Reports:

Method 1 - Backend-Centric With C Parsing

In The Initial Version, User Inputs Were Collected On The Frontend And Packaged Into A Structured Ji File Which Is The Jilang. This File Was Sent To The Backend Where A C Program Was Responsible For Parsing The Content And Converting It Into Latex Code. The Latex Was Then Compiled Into A Pdf Document. While Functional, This Method Had Limitations In Terms Of Scalability And Flexibility, Especially When It Came To Supporting Multiple Templates And Dynamic Content Formatting.

Method 2 - Frontend-Centric With Javascript Parsing

The Improved Version Moved The Latex Generation Logic To The Frontend Using Javascript. In This Method, The Latex Code Is Generated Directly In The Browser Based On User Input. This Code Is Then Sent To The Backend, Which Handles Pdf Compilation. This Approach Significantly Improves Performance, Reduces Backend Load, And Simplifies Scalability. A Major Advantage Of This Method Is The Support For Multiple Templates, Including A General Format And An Ieee-Compliant Format, Which Was Not Feasible In Method 1.

The Switch To A Frontend Javascript Parser Enabled The Use Of Customizable Templates. Users Can Now Choose Between A General Report Layout Or An Ieee-Compliant Format. The Parser Dynamically Formats The Content According To The Selected Template, Generating Latex That Reflects The Required Structure And Styling.

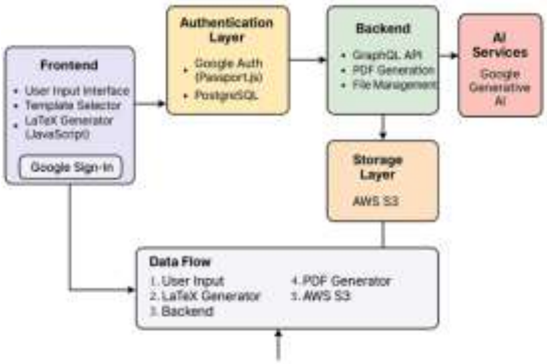


Fig 5: Architecture

As Shown In Figure 5, Reportease Is Designed With A Layered Architecture To Make The Report Generation Process Smooth And Efficient. The Frontend Is Built With Next.Js And Allows Users To Log In, Enter Content, And Choose Templates. Latex Code Is Generated Directly In The Browser Using Javascript, Which Is Faster And More Scalable Than The Older Backend Method.

Once The Latex Is Ready, It Is Sent To The Backend Built With Express.Js. The Backend Handles Pdf Creation And Stores The Final File Securely In Aws S3. Google Authentication Ensures Secure Access, And Graphql Is Used For Managing Data Efficiently. Ai Tools From Google Also Help Users With Smart Suggestions To Improve Their Reports.

Vi. Implementation

The Implementation Of The Report Generation Tool Involved Several Key Stages, Including Frontend Development, Backend Development, Integration With Aws Services And Google Ai Studio For Automated Report Generation.

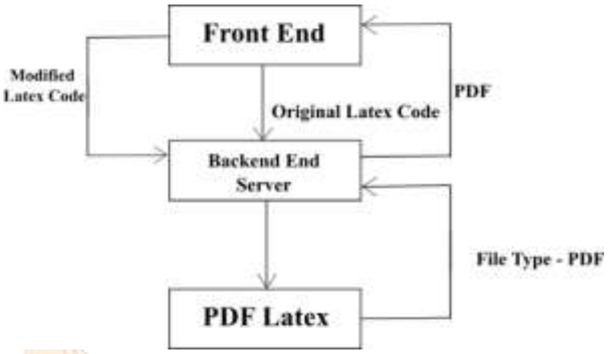


Fig 6: User Interaction

Figure 6 Indicates The Way Users Interact With Reportease To Construct And Organize Reports. The Process Begins As A User Logs In To The Site Frontend, Whereby They Can Input Their Content Into A Predefined Editor With Autosave. The Content Is Then Sent To The Backend Server Which Generates The Latex Code Based On The Content From Frontend. This Code Is Then Executed And Pdf Is Generated Which Can Be Modified By The User For Further Changes.

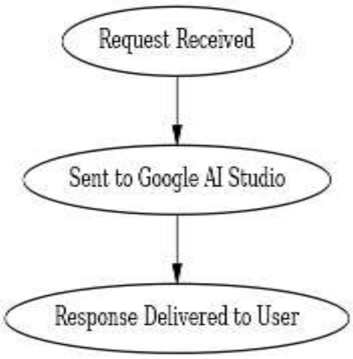
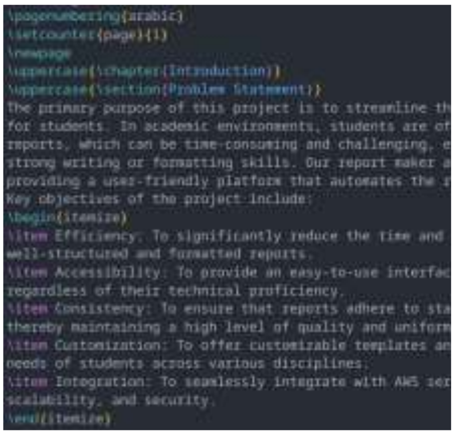


Fig 7: Request Flow

Figure 7 Illustrates The Process Of Processing User Inputs And Ai-Driven Support In Reportease. Once A User Sends A Request, It Is Received By The System First, Where It Is Processed. The Request Is Submitted To Google Ai Studio, Which Processes The Input And Produces An Ai-Driven Response. Last But Not Least, The Response Is Presented To The User, Offering Structured And High-Quality Report That Meets Academic Standards. This Streamlined Process Guarantees Effective Report Generation With Minimal Manual Effort.

Vii. Result

The Report Generation Tool Was Evaluated Based On Several Key Metrics, Including User Satisfaction, Report Quality, And System Performance.



```
\pagenumbering{arabic}
\setcounter{page}{1}
\newpage
\uppercase{\chapter{Introduction}}
\uppercase{\section{Problem Statement}}
The primary purpose of this project is to streamline th
for students. In academic environments, students are of
reports, which can be time-consuming and challenging, e
strong writing or formatting skills. Our report maker a
providing a user-friendly platform that automates the r
Key objectives of the project include:
\begin{itemize}
\item Efficiency: To significantly reduce the time and
well-structured and formatted reports.
\item Accessibility: To provide an easy-to-use interfac
regardless of their technical proficiency.
\item Consistency: To ensure that reports adhere to sta
thereby maintaining a high level of quality and uniform
\item Customization: To offer customizable templates an
needs of students across various disciplines.
\item Integration: To seamlessly integrate with AWS ser
scalability, and security.
\end{itemize}
```

Fig 8: An image of the code snippet from output.tex file

Figure 8 Shows The Final Output File Produced At The End Of The Process, Which Is Then Compiled And Used To Generate The Pdf.

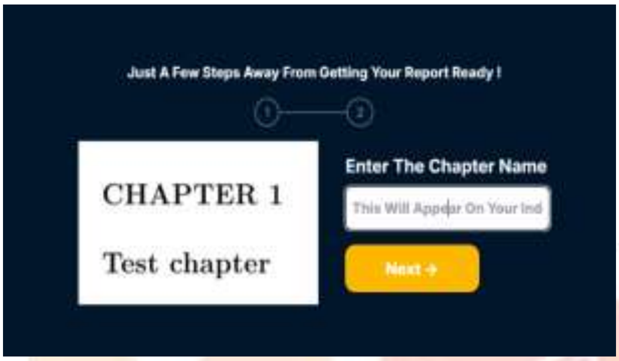

Just A Few Steps Away From Getting Your Report Ready !
1
2
CHAPTER 1
Test chapter
Enter The Chapter Name
This Will Appear On Your Ind
Next →

Fig 9: Chapter Name

The User Is First Asked To Input A Chapter Name Under Which He Or She Is Allowed To Add Content. Figure 9 Depicts The Ui The User Is Presented With.

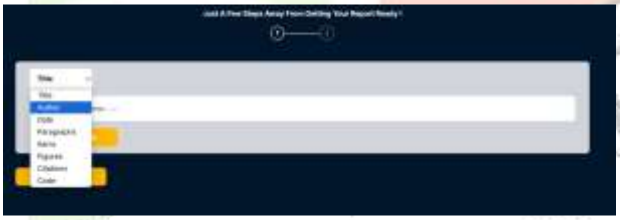


Fig 10: User Input

The User Is Then Redirected To The Next Page Where He Gets To Choose The Tags As Shown In Figure 10 According To The Requirements.

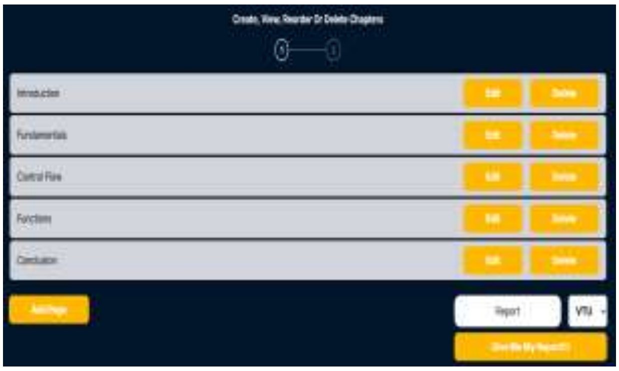
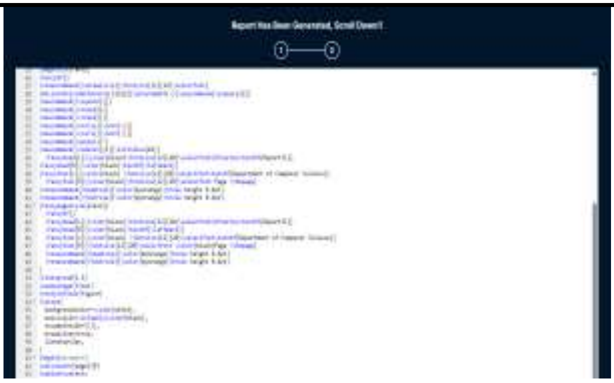


Fig 11: Listing Chapters

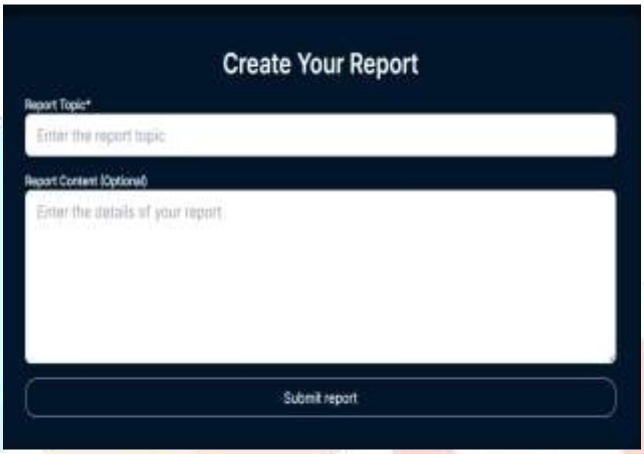
The User Than Gets Redirected To The Next Page Where He Can Review The Chapters And Choose What Template To Use Out Of Ieee And Generic Report Template. Figure 11 Illustrates This Page.



Report Has Been Generated, Scroll Down!

Fig 12: Latex Code

The User Is Presented With The Final Latex Code Generated By The Backend As Shown In Figure 12. The Report Pdf Will Be Generated Below The Code Snippet.



Create Your Report

Report Topic*

Enter the report topic

Report Content (Optional)

Enter the details of your report

Fig 13: Ai Report Generation

Figure 13 Shows The Ui The User Is Presented With When He Or She Wants A Framework Or A Point To Begin With A Report. It Uses Ai To Gather All The Content And Generate The Report In One Click Which Can Be Modified Further Accordingly.

Viii. Conclusion

Reportease Represents A Significant Advancement In Academic Report Generation, Offering Students A Reliable, Efficient, And Ai-Powered Platform For Creating High-Quality Reports With Minimal Effort. By Combining The Intuitive Usability Of Modern Word Processors With The Automation And Structured Formatting Of Latex, Reportease Simplifies The Complexities Of Academic Writing.

The Tool's Integration Of Ai-Driven Content Structuring, Realtime Formatting, And Cloud-Based Storage Ensures That Students Can Generate Well-Structured, Properly Formatted Reports While Focusing On Content Rather Than Manual Adjustments.

The System Also Emphasizes Scalability And Ease Of Use. Aws S3 Serves As A Secure, Centralized Repository For Storing And Retrieving Reports. Features Such As User Dashboards, Version Tracking, And Document Management Tools Enhance Productivity And Organization.

Ix. Future Scope

Reportease Has The Potential To Evolve Into A More Intelligent, Adaptable, And Comprehensive Academic Writing Tool. Future Enhancements Will Focus On Real-Time Collaboration, Enabling Multiple Users To Work On The Same Report Simultaneously While Maintaining Role-Based Access Control For Seamless Teamwork.

Additionally, Advanced Ai Assistance Will Be Improved To Refine Content Structuring, Provide Intelligent Citation Suggestions, And Integrate Grammar And Plagiarism Checks To Ensure Academic Integrity. To Further Enhance Usability, An Offline Mode Will Be Introduced, Allowing Users To Work On Reports Without An Internet Connection, While Mobile And Tablet Compatibility Will Ensure A Seamless Cross-Device Experience.

To Make The Platform Even More Versatile, Future Updates Will Also Focus On Improving Accessibility And User Personalization. Features Such As Customizable Templates, Dark Mode, And Accessibility-Friendly Themes Will Cater To A Broader Range Of Users With Diverse Preferences And Needs. These Enhancements Aim To Make Reportease Not Only A Powerful Academic Tool But Also An Inclusive And Usercentric Platform That Supports Productivity In Every Environment.

References

[1] O. Hartig And J. Pérez, “Graphql: A Systematic Mapping Study,” Proc. Acm Program. Lang., Vol. 6, No. Oopsla2, Pp. 1–26, 2022. Doi: 10.1145/3561818.

[2] M. R. Uddin, “Test Report Generation Using Json,” Researchgate, 2015. [Online]. Available: Https://Www.Researchgate.Net/Publication/281613871.

[3] P. Murray-Rust, R. Mounce, And M. P. Wilkerson, “Research Articles In Simplified Html: A Web-First Format For Html-Based Scholarly Articles,” Peerj Comput. Sci., Vol. 2, P. E132, 2016. Doi: 10.7717/Peerjcs.132.

[4] A. K. Mishra And R. Patel, “Gemini-The Most Powerful Llm: Myth Or Truth,” Ieee Xplore, 2024. [Online]. Available: Https://Ieeexplore.Ieee.Org/Document/10602253.

[5] A. P. Smith And B. Johnson, “Analyzing Next.Js's Effect On Search Engine Optimization And Website Speed (Seo),” Researchgate, 2023. [Online]. Available: Https://Www.Researchgate.Net/Publication/376046436.

[6] J. Williams And S. Brown, “Enhancing Seo In Single-Page Web Applications In Contrast With Multi-Page Applications,” Ieee Xplore, 2023. Doi: 10.1109/10403891.

[7] R. K. Sharma, “Essential Elements Of Writing A Research/Review Paper For Conference/Journals,” Ieee Xplore, 2018. Doi: 10.1109/8485210.

[8] M. Derntl, “Basics Of Research Paper Writing And Publishing,” Int. J. Tech. Enhanced Learning, Vol. 6, No. 2, Pp. 105–123, 2014.

[9] D. Peng, L. Cao, And W. Xu, "Using Json For Data Exchanging In Web Service Applications," *Journal Of Computational Information Systems*, Vol. 7, No. 16, Pp. 5883–5890, 2011.