E-CERTIFICATE GENERATOR

A MINI-PROJECT REPORT

Submitted By

MOHAMMED FARHAN (UNT22CS040)

MOHAMMED NOWFAL (UNT22CS042)

RADHESYAM RAGHAV K R (UNT22CS051)

SABAH K J (UNT22CS053)

To

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

in partial fulfillment for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

(SEMESTER VI)

2022-2026 Batch

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIVERSAL ENGINEERING COLLEGE

VALLIVATTOM, THRISSUR

MARCH 2025

DECLARATION

We undersigned hereby declare that the mini project report "E-CERTIFICATE GENERATOR",submitted for the partial fulfilment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under the supervision of our guide, Miss Jissmol Jose. – Assistant Professor,CSE department, UEC. This submission represents our idea in our own words and where ideas or words of others have been included; we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/ or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Vallivattom
29-03-2025

Signature:
Name of Student:.MOHAMMED FARHAN .

Signature:
Name of Student:.MOHAMMED NOWFAL .

Signature:
Name of Student:.RADHESYAM RAGHAV K R .

Signature:
Name of Student:.SABAH K J .

UNIVERSAL ENGINEERING COLLEGE
VALLIVATTOM, THRISSUR, 680123

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION

Empower universally equipped computational technocrats having innovative entrepreneurial skills with holistic values.

MISSION

- With well-experienced faculty members and excellent infrastructure, provide a way to equip universally acclaimed computational technocrats.
- With institute-industry interaction, develop apt skills in entrepreneurship and innovativeness.
- Provide value-based education to meet the technical and computational needs of society.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIVERSAL ENGINEERING COLLEGE

VALLIVATTOM, THRISSUR

BONAFIDE CERTIFICATE

This is to certify that the Mini-Project Report titled " **E-CERTIFICATE GENERATOR**"submitted by "**MOHAMMED FARHAN(UNT22CS040)**, **MOHAMMED NOWFAL(UNT22CS042)**, **RADHESYAM RAGAV K R(UNT22CS051)**, **SABAH K J(UNT22CS053)**" to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering is a bonafide record of the Mini-project work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose..

MINI-PROJECT GUIDE	**MINI-PROJECT CO-ORDINATOR**	**HEAD OF THE DEPT.**
Ms. Jissmol Jose	Prof. Dr. L.C.Manikandan	Ms. Nighila Ashok
Assistant Professor	Ms. Mahshiya V M , Asst.Professor	Associate Professor

Vallivattom

29/03/2025

ACKNOWLEDGMENT

We wish to record our indebtedness and thankfulness to all those who helped us to prepare this report titled **“E-certificate - Generator ”** and present it in a satisfactory way.

First and foremost, We thank God Almighty for His providence and for being the guiding light throughout the project.

We feel highly privileged in mentioning **Dr. Jose K Jacob , Principal, UEC** for his co-operation and help. We deeply and wholeheartedly thank **Ms. Nighila Ashok, HOD of CSE Department** for her extremely valuable advice and encouragement.

We are especially thankful to our coordinators **Dr. L.C.Manikandan, Professor, CSE** and **Ms. Mahshiya V M, Assistant Professor, CSE** for giving us valuable suggestions and critical inputs that helped us in completion of this project. We would also like to extend our heartfelt thanks to our guide **Ms. Jissmol Jose, Assistant Professor, CSE** for her meticulous guidance and support that helped us in the completion of our mini project.

We would like to extend our sincere gratitude to all faculty of Computer Science and Engineering department for the support and suggestions that helped us in the development of our project to what it is now. We thank our parents for the mental support provided during the course of the mini project at the times when our energies were the lowest.

ABSTRACT

In today's digital era, the need for efficient and professional certificate generation has become increasingly essential for organizations and individuals. To address this, we present the E-Certificate Generator, a comprehensive solution developed with Node.js for the backend and HTML/CSS/JavaScript for the user interface (UI). The challenge of tirelessly designing and distributing certificates is tackled by automating the process of creating customizable and professional certificates. Leveraging modern web technologies, the system allows users to design templates, upload participant data, and generate certificates in bulk or individually. Key features include a user-friendly template editor, support for multiple typefaces and colors, bulk certificate generation via CSV upload, and post-generation editing capabilities. By providing an intuitive interface and seamless functionality, our project aims to speed up the certificate creation process, empowering organizations to efficiently recognize achievements and enhance their professional branding.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API	Application Programming Interface
CSV	Comma-Separated Values
GUI	Graphical User Interface
PDF	Portable Document Format
UUID	Universally Unique Identifier
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
CSS	Cascading Style Sheets
JS	JavaScript
HTML	Hypertext Markup Language
PNG	Portable Network Graphics
TTF	TrueType Font
JSON	JavaScript Object Notation
SVG	Scalable Vector Graphics
SSL	Secure Sockets Layer
PDFKit	A Node.js library for generating PDFs
Multer	Middleware for handling file uploads in Node.js
UUIDv4	Version 4 of Universally Unique Identifier
CORS	Cross-Origin Resource Sharing

CHAPTER 1

INTRODUCTION

In today's digital era, the demand for efficient and professional recognition of achievements has grown significantly. Whether it's for academic accomplishments, corporate milestones, or event participation, certificates play a vital role in acknowledging and celebrating success. However, traditional methods of designing and distributing certificates often prove to be time-consuming and cumbersome. This challenge has paved the way for innovative solutions like the E-Certificate Generator, a tool designed to simplify and automate the certificate creation process

At its core, the E-Certificate Generator is a powerful application that combines modern web technologies with user-centric design to deliver a seamless certificate generation experience. By leveraging a robust backend built with Node.js and an intuitive frontend crafted with HTML, CSS, and JavaScript, this tool empowers users to create, customize, and distribute certificates effortlessly. From designing templates to generating certificates in bulk, the E-Certificate Generator addresses the diverse needs of organizations and individuals alike

Key features of the E-Certificate Generator include a dynamic template editor, support for bulk certificate generation via CSV uploads, and post-generation editing capabilities. Users can choose from a wide range of fonts, colors, and layouts to create visually appealing certificates that align with their branding. Additionally, the platform ensures accessibility and ease of use, making it suitable for users with varying levels of technical expertise.

Through meticulous development and a focus on user experience, the E-Certificate Generator aims to revolutionize the way certificates are created and distributed. This report explores the development process, features, and impact of the project, highlighting its significance in addressing the challenges of traditional certificate generation. By streamlining the process, the E-Certificate Generator not only saves time and effort but also enhances the professionalism and efficiency of certificate issuance in today's fast-paced world.

CHAPTER 2

LITERATURE SURVEY

In [1], the authors present a robust system for generating e-certificates using Node.js and a suite of powerful libraries, streamlining the creation and distribution process. This system leverages the Express.js framework for efficient server-side operations and routing. Multer middleware [2] facilitates seamless file uploads, particularly CSV files containing participant information. PDFKit [3] is employed to generate high-quality PDF certificates, ensuring professional-grade output. Additionally, Fabric.js [4] enables the creation and modification of canvas elements, allowing dynamic customization of certificate templates. Fast CSV [5] enhances performance through efficient parsing and formatting of CSV data, while UUID [6] ensures the production of unique identifiers for user sessions and file management.

The work discussed in [7] highlights the importance of a user-friendly interface for certificate generation. The system allows users to upload CSV files, select templates, and modify certificate details, ensuring an accessible and streamlined process. Tailwind CSS [7] enhances the visual appeal and responsiveness of the interface, contributing to a seamless user experience. Handlebars.js [8] is utilized for templating, enabling the dynamic rendering of certificate data and simplifying the customization process.

In [3], the focus is on optimizing certificate generation through the implementation of worker threads. By distributing workloads across multiple threads, the system efficiently handles large data volumes, ensuring timely certificate generation even under high load conditions. Robust error-handling mechanisms address issues related to file uploads and processing errors. Additionally, session management safeguards user-specific data, maintaining security throughout the certificate generation process.

The system also incorporates Archiver [9], which provides a streaming interface for generating compressed archives of certificates, enabling bulk downloads for users. This feature is particularly beneficial for organizations managing large-scale events. Furthermore, the use of Node.js [10] as the runtime environment ensures cross-platform compatibility and scalability, allowing the system to handle diverse deployment scenarios effectively.

CHAPTER 3

EXISTING SYSTEM

3.1 INTRODUCTION

The existing system for certificate generation relies on manual processes, requiring the use of software like Microsoft Word, PowerPoint, or graphic design tools. Users must manually input participant details, adjust layouts, and format certificates, which is both time-consuming and prone to errors such as incorrect names or event details.

In many cases, certificates are printed and physically distributed, incurring additional costs for printing, logistics, and storage. When issued digitally, users must manually convert designs into PDFs, a repetitive and inefficient task. The lack of a centralized system for managing templates, participant data, and issued certificates results in inconsistencies and difficulties in maintaining records.

The existing system does not support automation or advanced features such as bulk certificate generation, real-time customization, or post-generation editing. Users must depend on multiple tools for designing, generating, and distributing certificates, leading to fragmented workflows. Additionally, retrieving past certificates is often cumbersome due to the absence of centralized storage, making verification and record-keeping challenging

Overall, the existing system is inefficient, outdated, and unsuitable for modern organizations that require a scalable, automated, and user-friendly certificate generation process.

3.2 METHODOLOGY

The traditional certificate generation process follows a structured but highly manual approach. Initially, users create certificate templates using design software, where static layouts require manual modifications for each recipient. Participant details such as names, event titles, and dates are entered manually or imported from spreadsheets, necessitating additional tools for merging data with templates.

For large-scale events, users often manage participant information using spreadsheets. However, integrating this data with certificates requires either manual intervention or third-party tools, increasing the likelihood of errors. Once generated, certificates are either printed or converted into PDFs for distribution via email or other digital channels. This workflow is inefficient, particularly for events with a large number of participants.

Furthermore, there is no integration between the design, data management, and distribution processes, leading to fragmented workflows. The absence of automation means that users must manually verify details for accuracy, further extending the time required to complete the task. Additionally, without a centralized storage system, organizing and retrieving past certificates becomes difficult.

The methodology of the existing system highlights the need for an improved approach that integrates automation, centralized data management, and streamlined distribution to enhance efficiency and accuracy.

3.3 ADVANTAGES

- Customizable Templates: Users can design certificates with personalized layouts, fonts, and colors.
- Familiar Tools: The use of widely available software ensures accessibility and ease of use.
- Tangible Recognition: Physical certificates provide recipients with a formal record of achievement.

3.4 DISADVANTAGES

- Time-Consuming: Manual entry and formatting require significant effort, particularly for bulk certificate generation.
- Prone to Errors: Mistakes such as incorrect names or event details are common due to manual data entry.
- Lack of Automation: The absence of automated processes limits efficiency and scalability.
- High Costs: Printing and distributing physical certificates incur expenses for materials and logistics.
- Fragmented Workflow: Users rely on multiple tools, making the process inefficient and difficult to manage.
- No Centralized Storage: Certificates are not systematically stored, making retrieval and verification difficult.
- Limited Scalability: The system struggles to handle large events due to its manual nature.

CHAPTER 4

PROPOSED SYSTEM

4.1 INTRODUCTION

The proposed system for the E-Certificate Generator project aims to automate and simplify the process of designing, generating, and distributing professional certificates. It addresses the inefficiencies of traditional methods by introducing advanced features and technologies to enhance productivity, accuracy, and scalability.

The system includes a dynamic certificate template editor that allows users to create customized designs with features like drag-and-drop functionality, font and color customization, and the ability to upload logos and signatures. Pre-designed templates are also available for quick customization.

A key feature is bulk certificate generation, which processes participant data from CSV files to generate multiple certificates simultaneously.

Post-editing capabilities allow users to make real-time adjustments to certificates after generation. This ensures flexibility and reduces errors, with features like version control and error correction.Proper formatting and localization ensure accuracy for diverse linguistic needs.

A centralized certificate management dashboard provides tools for previewing, downloading, and organizing certificates. Users can download certificates individually or in bulk as ZIP files, search and filter certificates, and share them via email or download links.

The system also includes an analytics dashboard to track metrics such as the number of certificates generated, participant engagement, and template usage. A feedback mechanism allows users to provide suggestions, enabling continuous improvement of the system.

In conclusion, the proposed E-Certificate Generator system is a comprehensive and user-friendly solution designed to streamline certificate creation and management. It leverages advanced technologies to deliver efficiency, customization, and scalability for organizations and individuals.

4.2 METHODOLOGY

A comprehensive approach was adopted to streamline the process of generating professional e-certificates efficiently and accurately. In Figure 4.1, the block diagram of this methodology illustrates the sequential phases involved in the e-certificate generation process.

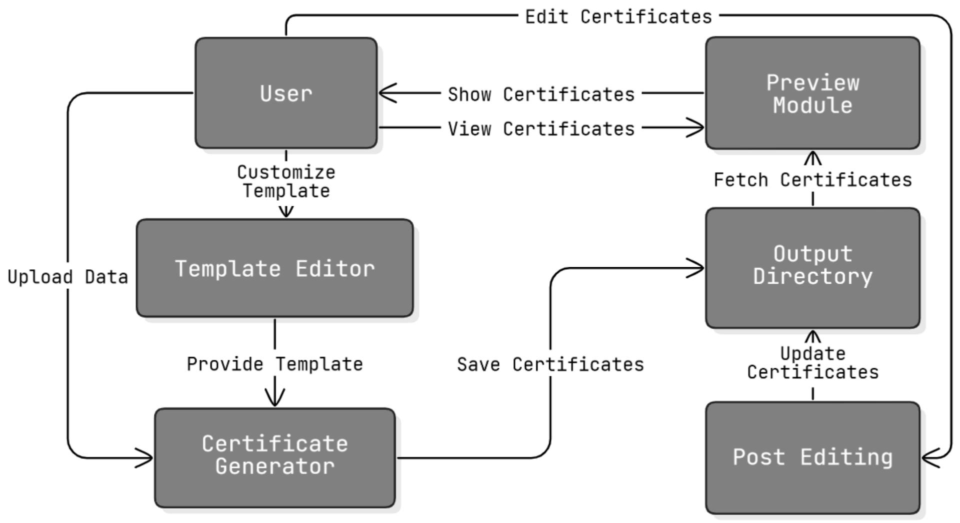


Fig. 4.1: Block diagram of the proposed methodology.

The project began with a detailed analysis of user requirements and preferences through surveys, interviews, and market research. This phase aimed to understand the challenges users face in creating and distributing certificates for events, workshops, and other occasions. By gathering feedback from potential users, the team identified the essential features and functionalities required for an efficient e-certificate generation tool.

In this phase, the system was designed to accept user-provided data in the form of CSV files containing participant details such as names and event descriptions.refered from [5]. The system also allowed users to customize provided certificate templates. Preprocessing techniques were implemented to validate and clean the input data, ensuring that errors such as missing fields or invalid formats were handled gracefully.

A library of professionally designed certificate templates was created to cater to diverse user needs. Each template was configured with predefined coordinates for text placement, font styles, and colors. The system also allowed users to customize templates using the integrated Template Editor. refered from [4] , which provided features such as adding text, images, and adjusting design elements. Advanced PDF generation techniques were employed to create high-quality certificates. The system utilized the PDFKit library to dynamically generate certificates based on the selected template and user-provided data.This was refered from [3] . Key features included:

•Accurate placement of participant names and event details on the certificate.

•Support for multiple fonts and colors to match the design aesthetics.

•Batch processing for generating certificates in bulk using CSV data.

An intuitive and user-friendly interface was developed to facilitate seamless interaction with the system. refered from [7]. The web-based interface allowed users to:

•Upload CSV files and customize template.

•Generate certificates individually or in bulk.

•Perview and Post-Generation error fixing

•Download generated certificates or export them as a ZIP file.

The design prioritized usability and accessibility, ensuring that users could easily navigate the system and complete their tasks efficiently.

The backend algorithms, user interface, and middleware components were integrated to create a cohesive system. refered from [1]. Rigorous testing was conducted to evaluate the functionality, performance, and usability of the system across different devices and browsers. This phase ensured that the system met the project requirements and provided a reliable solution for users.

After deployment, the system underwent continuous monitoring to track performance and gather user feedback. Regular updates and improvements were implemented based on user needs and technological advancements. This ensured that the system remained effective and up-to-date in addressing the evolving challenges of certificate generation.

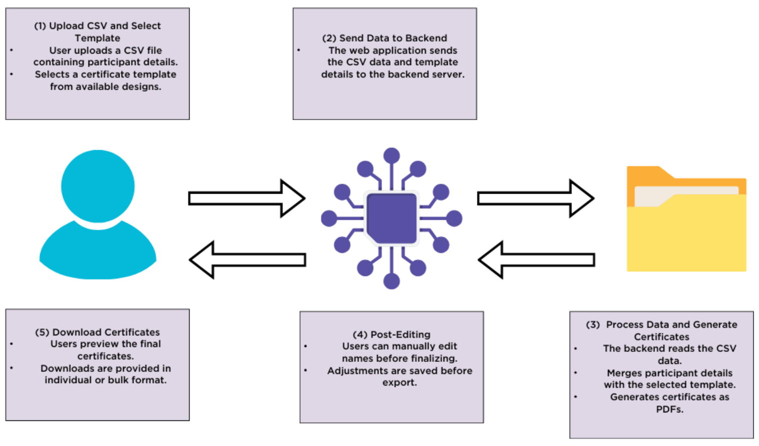

(1) Upload CSV and Select Template
• User uploads a CSV file containing participant details.
• Selects a certificate template from available designs.
(2) Send Data to Backend
• The web application sends the CSV data and template details to the backend server.
(5) Download Certificates
• Users preview the final certificates.
• Downloads are provided in individual or bulk format.
(4) Post-Editing
• Users can manually edit names before finalizing.
• Adjustments are saved before export.
(3) Process Data and Generate Certificates
• The backend reads the CSV data.
• Merges participant details with the selected template.
• Generates certificates as PDFs.

Fig. 4.2: Proposed Working

4.3 ADVANTAGES

- Cost-Effective and Accessible: The E-Certificate Generator provides an affordable and efficient solution for certificate creation, making it accessible to organizations and individuals with limited budgets. It eliminates the need for expensive manual processes.
- Time-Saving: Automates the certificate generation process, significantly reducing the time required to create and distribute certificates, especially for large-scale events.
- Customizable Designs: Offers a dynamic template editor with advanced customization options, allowing users to create professional and personalized certificates that align with their branding.
- Bulk Certificate Generation: Supports the generation of multiple certificates simultaneously using CSV uploads, streamlining the process for events with numerous participants.
- Real-Time Editing: Allows users to make adjustments to certificates even after generation, ensuring flexibility and reducing errors.

4.4 DISADVANTAGES

- Technical Complexity: Setting up and using the system may require a learning curve, especially for users unfamiliar with advanced design tools or bulk data processing.
- Dependency on Internet Connectivity: The system relies on internet access for certain features, such as cloud storage and font fetching, which may be a limitation in areas with poor connectivity.
- Customization Limitations: Although the system offers extensive customization options, users with highly specific design requirements may find the tools insufficient compared to professional graphic design software.
- Data Privacy Concerns: Handling participant data for bulk certificate generation may raise privacy concerns, requiring strict compliance with data protection regulations.

CHAPTER 5

SOFTWARE REQUIREMENT SPECIFICATION

5.1 INTRODUCTION

5.1.1 Purpose

The purpose of the E-Certificate Generator project is to streamline the process of designing, generating, and distributing professional certificates. By leveraging modern web technologies, the project aims to automate certificate creation, allowing users to generate single or bulk certificates with ease. This tool is designed to cater to the needs of organizations, event managers, and educational institutions, providing a time-efficient and user-friendly solution for certificate generation. The ultimate goal is to enhance productivity, reduce manual effort, and ensure consistency in certificate design and distribution.

5.1.2 Document Conventions

This document follows the IEEE format standard (IEEE Std. 830-1998).

5.1.3 Intended Audience and Reading Suggestions

This Software Requirements Specification (SRS) document is intended for developers, project managers, stakeholders, and quality assurance teams involved in the E-Certificate Generator project. Developers can refer to the technical specifications, project managers and stakeholders can gain insights into the project's goals and constraints, and quality assurance teams can use it as a guide for testing. For an efficient understanding, start with the introduction, explore functional and non-functional requirements, and review system architecture, user stories, and interface details. Sequential navigation ensures a comprehensive understanding and facilitates collaboration among all stakeholders.

5.1.4 Project Scope

The E-Certificate Generator is a web-based application designed to simplify the process of certificate creation. It allows users to choose templates, upload participant data, and generate certificates in bulk or individually. The system supports customizable templates, real-time previews, and post-generation editing. By automating repetitive tasks, the project aims to save time and effort while ensuring accuracy and professionalism in certificate issuance. This scope aligns with the broader vision of providing a scalable and efficient solution for certificate generation across various domains.

5.2 OVERALL DESCRIPTION

5.2.1 Product Perspective

The E-Certificate Generator is a standalone web application designed to address the inefficiencies of traditional certificate generation methods. Unlike existing manual systems, it automates the entire process, from template design to certificate distribution. The system integrates seamlessly with modern web technologies and supports CSV-based bulk data uploads for large-scale events. It operates independently but can be extended to integrate with third-party tools for additional functionalities. A high-level architecture diagram is included in the project documentation to illustrate the system's components and their interactions.

5.2.2 Product Features

The E-Certificate Generator offers the following key features:

Customizable Templates:

Users can choose and customize templates with custom text, colors, and layouts.

Bulk Certificate Generation:

Supports CSV uploads for generating certificates in bulk.

Real-Time Preview:

Allows users to preview certificates before finalizing them.

Post-Generation Editing:

Enables users to make corrections or updates to generated certificates.

High-Resolution Exports:

Certificates can be downloaded in high-quality PDF format.

These features collectively provide a comprehensive solution for certificate generation, ensuring efficiency, accuracy, and professionalism.

5.2.3 User Classes and Characteristics

The E-Certificate Generator is designed to cater to the following user classes:

Event Organizers:

- Frequency: Regular users generating certificates for events.
- Characteristics: Limited technical expertise, seeking a simple and efficient solution.

Educational Institutions:

- Frequency: Frequent users issuing certificates for courses and workshop.
- Characteristics: Moderate technical expertise, requiring bulk generation capabilities.

Corporate Users:

- Frequency: Occasional users for employee recognition or training programs.
- Characteristics: High focus on branding and customization.

Administrators:

- Frequency: Infrequent users managing system configurations.
- Characteristics: High technical expertise, responsible for system maintenance.

The system ensures a user-friendly experience for all user classes while providing advanced functionalities for power users.

5.2.4 Operating Environment

The E-Certificate Generator operates in the following environment:

Hardware Platform:

Compatible with standard hardware configurations, including desktops and laptops.

Operating System:

Supports Windows, macOS, and Linux distributions.

Web Browsers:

Compatible with modern browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge.

Web Technologies:

Built using Node.js, Express.js, HTML, CSS, and JavaScript.

API Integration:

Utilizes libraries like PDFKit and Fabric.js for PDF generation and canvas manipulation. This broad compatibility ensures accessibility and ease of use for a wide range of users.

5.2.5 Design and Implementation Constraints

The design and implementation of the E-Certificate Generator are guided by several key constraints to ensure a robust and reliable system. Data privacy is prioritized by securely handling user data and temporary files. Performance optimization is achieved to efficiently process large datasets during bulk certificate generation. The project strictly adheres to the chosen technology stack of Node.js [10], Express.js [1], and related libraries. File size limitations are enforced, restricting uploads to a maximum of 10 MB. Additionally, session-based user identification is implemented to provide a secure and personalized experience. These constraints collectively contribute to the system's efficiency, security, and usability.

5.2.6 User Documentation

The E-Certificate Generator will feature comprehensive user documentation to ensure a seamless experience. It will include step-by-step instructions for designing templates and generating certificates, along with FAQs and troubleshooting guides to assist users in resolving common issues. Additionally, online help features will be integrated into the web platform for easy access. The documentation will be available in digital formats, ensuring convenient navigation and accessibility for all users.

5.2.7 Assumptions and Dependencies

The development of the E-Certificate Generator is based on several key assumptions and dependencies to ensure smooth functionality. It relies on third-party libraries like PDFKit and Fabric.js for core features such as template design and PDF generation. A stable development environment is assumed, ensuring the reliability of the Node.js runtime and associated frameworks. The system also depends on the accuracy of user input for the correct generation of certificates. In addition, browser compatibility is a crucial factor, relying on the consistent behavior of modern web browsers to deliver a seamless user experience.

5.3 SYSTEM FEATURES

5.3.1 Template Editor

Description and Priority:

Description: Allows users to customize certificate templates by adding text, images, and adjusting design elements.
Priority: High

Stimulus/Response Sequences:

Stimulus: User selects a template and customizes it using the editor. Response: System saves the customized template for future use.

Functional Requirements:

The system must allow users to upload images and add text to templates.

The system must provide options to select fonts, colors, and positions for text elements.

5.3.2 Single Certificate Generation

Description and Priority:

Description: Enables users to generate a single certificate by entering participant details . Priority: Medium

Stimulus/Response Sequences:

stimulus: User provides participant name and event details. Response: System generates a certificate in PDF format and allows the user to download it.

Functional Requirements:

The system must generate a certificate based on the selected template and user-provided details.

The system must allow users to download the generated certificate.

5.3.3 Bulk Certificate Generation

Description and Priority:

Description: Allows users to generate multiple certificates by uploading a CSV file containing participant details.
Priority: High

Stimulus/Response Sequences:

Stimulus: User uploads a CSV file and provides event details. Response: System processes the file and generates certificates for all participants.

Functional Requirements:

The system must validate the uploaded CSV file and extract participant details. The system must generate certificates in bulk and provide a ZIP file for download.

5.3.4 Certificate Preview and Editing

Description and Priority:

Description: Enables users to preview generated certificates and make corrections if needed.
Priority: Medium

Stimulus/Response Sequences:

Stimulus: User selects a certificate to preview or edit. Response: System displays the certificate and allows modifications.

Functional Requirements:

The system must allow users to preview certificates in a browser. The system must enable users to edit participant details and regenerate the certificate.

5.4 EXTERNAL INTERFACE REQUIREMENTS

5.4.1 User Interfaces

- Template Editor Interface: A user-friendly interface for selecting and customizing certificate templates. Includes dropdowns for fonts, color pickers, and options to upload images or add text. Provides a live preview of the certificate.
- Single Certificate Interface: A simple form for entering participant details (e.g., name and event description) and generating a single certificate. Includes a preview section for the generated certificate.
- Summarization Interface: Real-time feedback during summarization with progress indicators. Clear display of the finalized summary in a user-friendly format.
- Bulk Certificate Interface: Allows users to upload a CSV file and enter default event details. Displays progress during certificate generation and provides a link to preview or download the certificates.
- Preview Interface: Displays generated certificates in a grid format with options to download individual certificates, fix details, or download all as a ZIP file. Includes a search bar for filtering certificates.
- Error Handling Interface: Standardized error messages displayed in a clean format. Provides clear feedback for issues like invalid file uploads or missing participant details.

5.4.2 Hardware Interfaces

- System Requirements: The extension should be compatible with standard hardware configurations, such as desktops, laptops, and compatible web browsers.
- Output Devices: Certificates are generated as PDF files, compatible with standard printers and display devices.
- Storage requirements: Temporary storage is required for uploaded files (e.g., CSV files and templates) and generated certificates.

5.4.3 Software Interfaces

PDFKit Library:

Used for generating PDF certificates with custom fonts, colors, and layouts.

Fast-CSV Library:

Processes CSV files for extracting participant details.

Express.js Framework:

Handles routing and server-side logic for certificate generation and template editing.

Fabric.js Library:

Provides a canvas-based editor for customizing certificate templates.

Web Browsers:

The application is compatible with modern web browsers like Chrome, Firefox, and Edge.

File System:

Interacts with the file system for storing uploaded templates, CSV files, and generated certificates.

5.4.4 Communications Interfaces

HTTP/HTTPS Protocols:

The application uses HTTP/HTTPS protocols for communication between the frontend and backend components.

Data Transfer:

Data such as participant details (from CSV files) and generated certificates are transferred securely between components.

Message Formatting:

JSON is used for structured communication between the frontend and backend.

Security Measures:

HTTPS ensures secure data transfer, and session management prevents unauthorized access to user-specific files.

Synchronization Mechanisms:

Synchronization between the template editor, certificate generator, and preview components ensures consistent data processing and real-time updates.

5.5 OTHER NONFUNCTIONAL REQUIREMENTS

5.5.1 Performance Requirements

Response Time:

The system should generate an e-certificate within a maximum response time of 10 seconds after the user submits the request. This ensures an efficient and responsive user experience.

Processing Time:

The backend processing of certificate generation should not exceed 20 seconds per batch of 50 certificates. This guarantees timely certificate delivery, even for bulk requests.

Concurrency Support:

The system should handle concurrent certificate generation requests from multiple users without performance degradation.

Compatibility Across Browsers:

The web-based interface should function consistently across major browsers like Chrome, Firefox, and Edge, ensuring accessibility for all users.

Scalability:

The system should efficiently handle a growing number of users and certificate requests without significant performance drops.

Resource Utilization:

The application should be optimized for efficient resource usage, ensuring minimal CPU and memory consumption during certificate generation.

Network Latency:

The system should minimize the impact of network delays by optimizing API interactions and certificate delivery mechanisms.

Real-Time Interaction:

For real-time generation, user interactions and certificate processing updates should occur with minimal delay, maintaining a seamless user experience.

5.5.2 Safety Requirements

User Data Protection:

User data, including names and certificate details, should be encrypted during storage and transmission to prevent unauthorized access.

Content Accuracy:

Ensure the accuracy of generated certificates to prevent incorrect information from being issued.

Prevent Overload:

Implement rate-limiting and load-balancing mechanisms to prevent server overload during peak usage.

User Focus:

The interface should be intuitive and distraction-free to minimize errors while submitting certificate details.

Policy Compliance:

The system should comply with data privacy regulations and best practices for digital certification.

Certifications:

Obtain and maintain relevant security and reliability certifications to ensure trustworthiness.

Data Integrity:

Ensure that generated certificates remain unaltered and verifiable, preventing unauthorized modifications.

5.5.3 Security Requirements

User Authentication:

Implement secure authentication to restrict access to authorized personnel only.

Vulnerability Scanning:

Regularly perform security scans to identify and fix vulnerabilities.

Logging and Monitoring:

Maintain logs and real-time monitoring to detect and respond to suspicious activities.

Privacy Compliance:

Adhere to data protection regulations to safeguard user information.

Security Certifications:

Ensure compliance with industry-standard security certifications for data protection.

5.5.4 Software Quality Attributes

Usability:

Provide an intuitive and easy-to-use interface, targeting a System Usability Scale (SUS) score of at least 80.

Reliability:

Ensure a 99

Maintainability:

Maintain a structured and well-documented codebase to support future updates and improvements.

Flexibility:

Allow customization of certificate templates to adapt to different organizational needs.

Interoperability:

Ensure seamless integration with third-party systems, such as email services for certificate distribution.

Adaptability:

Ensure the system remains compatible with evolving web and security standards.

Portability:

Design the system to be deployable on multiple platforms, including cloud and on-premise servers.

Correctness:

Use automated validation to ensure the correctness of generated certificates.

5.5.5 Other Requirements

Legal Compliance:

Ensure adherence to copyright laws and regulations related to digital certificates.

Reuse Objectives:

Encourage modular code development for easy reuse and future feature expansions.

Error Handling:

Implement clear and informative error messages to guide users when issues occur.

Performance Monitoring:

Incorporate real-time performance tracking to optimize system efficiency.

User Feedback Mechanism:

Provide a feedback system for users to report issues and suggest improvements.

5.6 GLOSSARY

CSV:

Comma-Separated Values, a file format used for storing tabular data such as participant details.

PDF:

Portable Document Format, a file format used for presenting certificates in a printable and shareable format.

Template:

A pre-designed layout used as a base for generating certificates.

Bulk Certificate Generation:

The process of creating multiple certificates at once using a CSV file containing participant details.

Frontend:

The client-side of the application responsible for user interface and interactions, such as the certificate editor and preview screens.

Backend:

The server-side of the application responsible for processing logic, such as generating certificates and handling file uploads.

Fabric.js:

A JavaScript library used for creating and editing certificate templates on a canvas.

PDFKit:

A JavaScript library used for generating PDF files programmatically.

Fast-CSV:

A Node.js library used for parsing and processing CSV files.

HTTPS:

Hypertext Transfer Protocol Secure, a secure version of HTTP used for secure communication between the client and server.

ZIP File:

A compressed file format used for bundling multiple certificates into a single downloadable file.

Error Handling:

The process of identifying, managing, and resolving errors in the application, such as invalid file uploads or missing data.

Certificate Preview:

A feature that allows users to view generated certificates before downloading or printing them.

Certificate Preview:

A feature that allows users to view generated certificates before downloading or printing them.

Customization:

The process of modifying certificate templates by adding text, images, and adjusting design elements.

5.7 ANALYSIS MODELS

5.7.1 Use Case Diagram

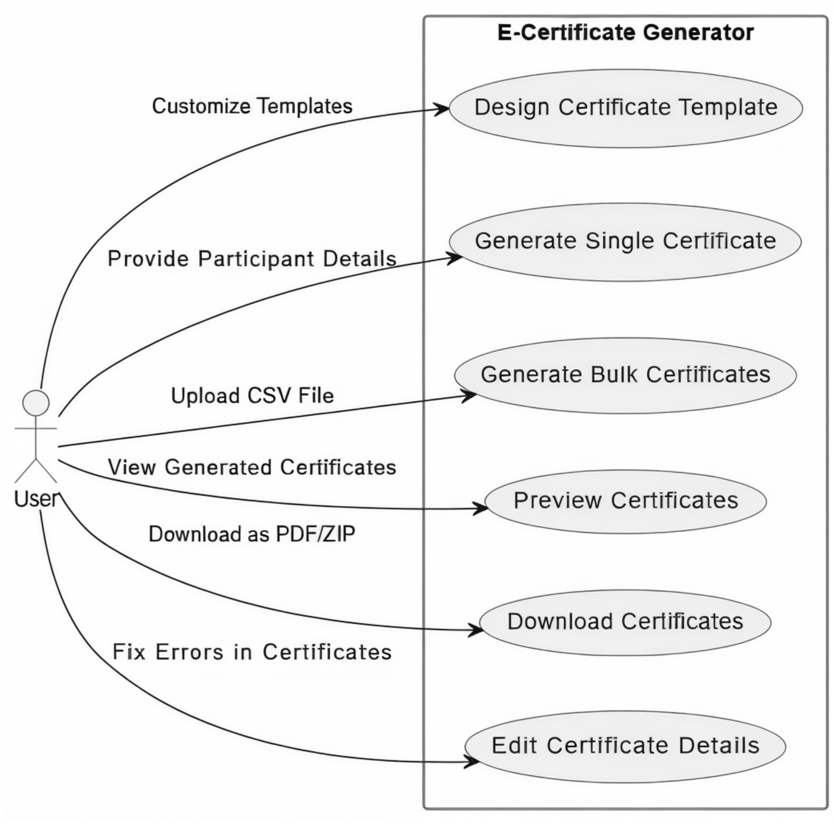

E-Certificate Generator
Customize Templates
Design Certificate Template
Provide Participant Details
Generate Single Certificate
Upload CSV File
Generate Bulk Certificates
View Generated Certificates
Preview Certificates
User
Download as PDF/ZIP
Download Certificates
Fix Errors in Certificates
Edit Certificate Details

Fig. 5.1: Use Case Diagram

5.7.2 Data Flow Diagram

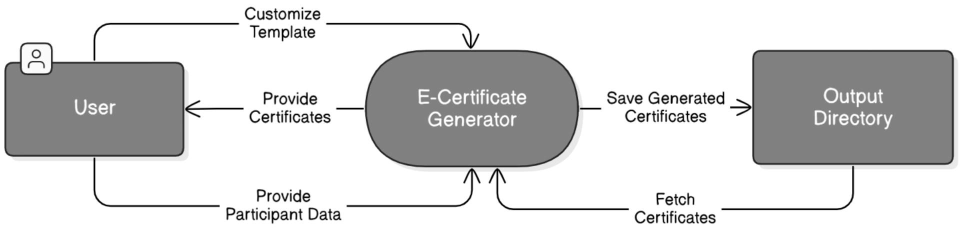


Fig. 5.2: DFD-Level 0

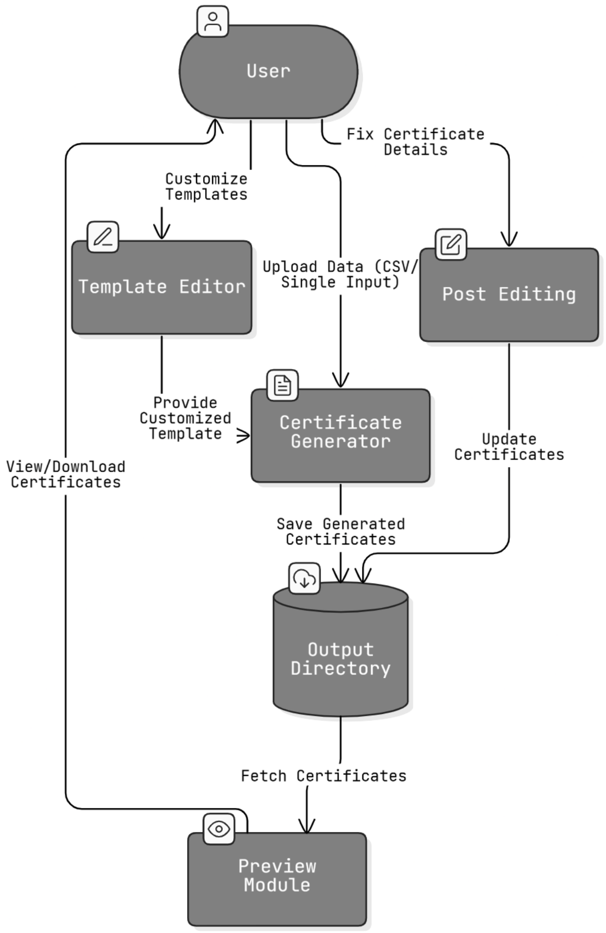


Fig. 5.3: DFD-Level 1

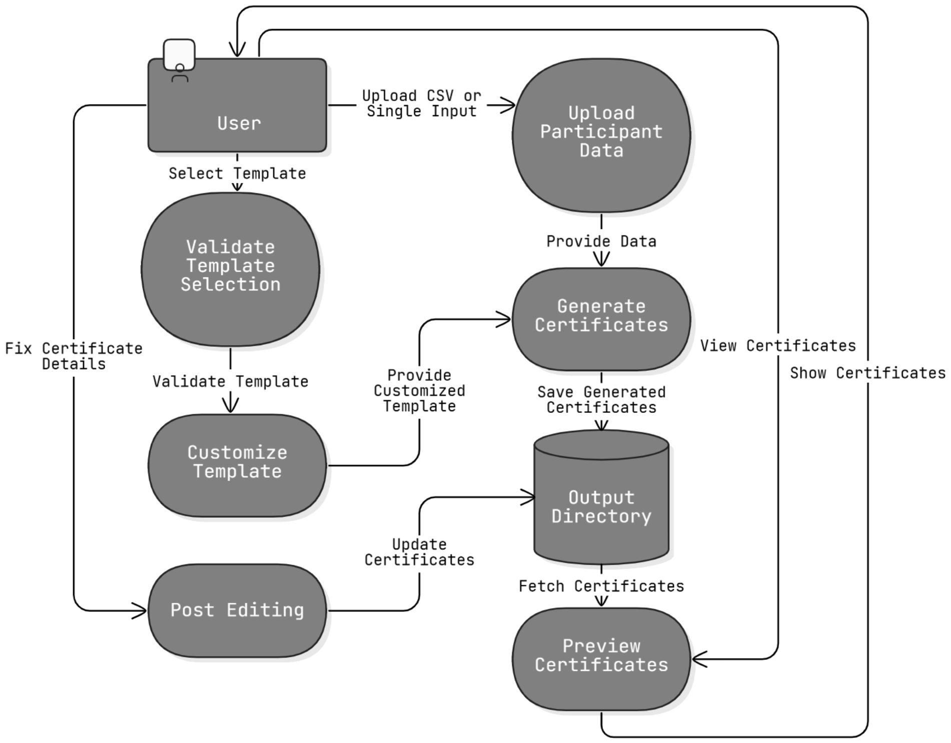


Fig. 5.4: DFD-Level 2

5.7.3 Class Diagram

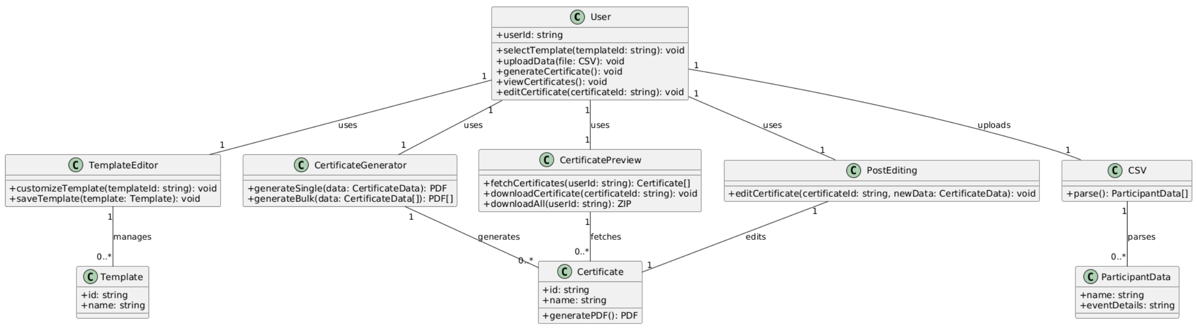

User
+userId: string
+selectTemplate(templateId: string): void
+uploadData(file: CSV): void
+generateCertificate(): void
+viewCertificates(): void
+editCertificate(certificateId: string): void
TemplateEditor
+customizeTemplate(templateId: string): void
+saveTemplate(template: Template): void
CertificateGenerator
+generateSingle(data: CertificateData): PDF
+generateBulk(data: CertificateData[]): PDF[]
CertificatePreview
+fetchCertificates(userId: string): Certificate[]
+downloadCertificate(certificateId: string): void
+downloadAll(userId: string): ZIP
PostEditing
+editCertificate(certificateId: string, newData: CertificateData): void
CSV
+parse(): ParticipantData[]
Template
+id: string
+name: string
Certificate
+id: string
+name: string
+generatePDF(): PDF
ParticipantData
+name: string
+eventDetails: string
uses
uses
uses
uses
uploads
manages
generates
fetches
edits
parses
1
0..*

Fig. 5.5: Class Diagram

5.8 ISSUES LIST

5.8.1 TBD - Certificate Generation

- **Decision Pending**
  - **Template Customization:** Determine additional customization options for certificate templates.
- **Info Needed**
  - **Font Licensing:** Verify licensing requirements for bundled fonts.
  - **PDFKit Limitations:** Research limitations of PDFKit for advanced certificate designs.

5.8.2 TBD - User Experience

- **Decision Pending**
  - **Post-Generation Editing:** Define implementation details for editing certificates after generation.
- **Info Needed**
  - **Browser Compatibility:** Decide on extending compatibility to additional web browsers.
  - **Mobile Optimization:** Gather specific requirements for optimizing the application for mobile devices.

5.8.3 TBD - Performance

- **Decision Pending**
  - **Bulk Generation:** Decide on strategies to optimize bulk certificate generation for large datasets.
- **Info Needed**
  - **Worker Threads:** Evaluate the use of worker threads for parallel processing.
  - **Server Load Testing:** Conduct load testing to determine server capacity for concurrent users.

5.8.4 TBD - Error Handling

- **Decision Pending**
  - **Error Logging:** Decide on strategies for centralized error logging and monitoring.
- **Info Needed**

  - **User Notifications:** Gather requirements for notifying users about errors during certificate generation.

5.8.5 TBD - Internationalization

- **Decision Pending**
  - **Language Support:** Define the scope of languages to be supported for certificate text.
- **Info Needed**
  - **Font Compatibility:** Verify font compatibility for non-Latin scripts.
  - **Translation Workflow:** Determine the workflow for managing translations.

CHAPTER 6

SOFTWARE DESIGN DOCUMENT

6.1 INTRODUCTION

6.1.1 Purpose

This Software Design Document (SDD) outlines the detailed architecture and system design for the ”E-Certificate Generator” project. It describes the technical components, their functionalities, and how they interact to achieve the overall goal of automating the design, generation, and distribution of professional certificates. This SDD serves as a blueprint for developers to understand the technical design decisions and implement the functionalities of the system.

6.1.2 Scope

The ”E-Certificate Generator” is a web-based application designed to simplify and automate the process of certificate creation. It allows users to select and view templates, upload participant data, and generate certificates either individually or in bulk. The system supports features such as customizable templates, and post-generation editing. By automating repetitive tasks, the project aims to save time and effort while ensuring accuracy and professionalism in certificate issuance. The scope of this project aligns with the broader vision of providing a scalable, efficient, and accessible solution for certificate generation across various domains.

6.1.3 Overview

This Software Design Document (SDD) provides a comprehensive technical blueprint for the ”E-Certificate Generator” project. It outlines the functionalities (automating certificate creation and distribution), goals (enhancing productivity and reducing manual effort), and benefits (improved efficiency and accessibility) in the Scope section. The following sections delve into the system architecture (component interactions and design approach), provide detailed design of individual functionalities, outline the testing strategy, and conclude with future considerations, all serving as a comprehensive guide for developers and stakeholders.

6.2 SYSTEM OVERVIEW

The E-Certificate Generator is a comprehensive software solution developed to efficiently automate the process of certificate creation and distribution. Consisting of a user-friendly web platform, the system allows users to view and select templates, upload participant data, and generate certificates with ease. This project operates within the broader context of event management and professional certification, focusing specifically on enhancing the efficiency and accuracy of certificate generation. The system's design prioritizes simplicity, scalability, and accessibility to provide a seamless and intuitive user experience. Motivated by the challenges associated with manual certificate creation, the E-Certificate Generator addresses the evolving needs of users, offering a tool that automates repetitive tasks without compromising quality or professionalism.

6.3 SYSTEM ARCHITECTURE

6.3.1 Architectural Design

In this architectural design, users interact with the system through a web-based interface, which facilitates access to the certificate generation functionality. The system allows users to upload participant data via CSV files [2] and select certificate templates [4]. The uploaded data and selected templates are processed by the backend module, which generates certificates using the PDF generation engine [3]. The generated certificates are stored temporarily and made available for preview, download, or bulk export as a ZIP file [9]. This design provides a simplified overview of the main components and interactions involved in the E-Certificate Generator system.

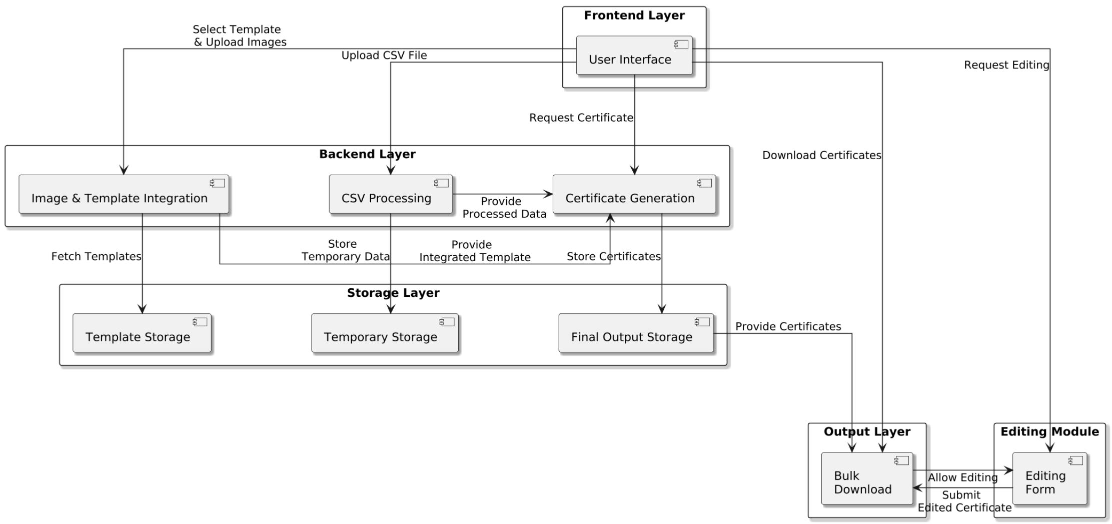


Fig. 6.1: Architectural Diagram of Proposed System

6.3.2 Decomposition Description

Sequence Diagram:

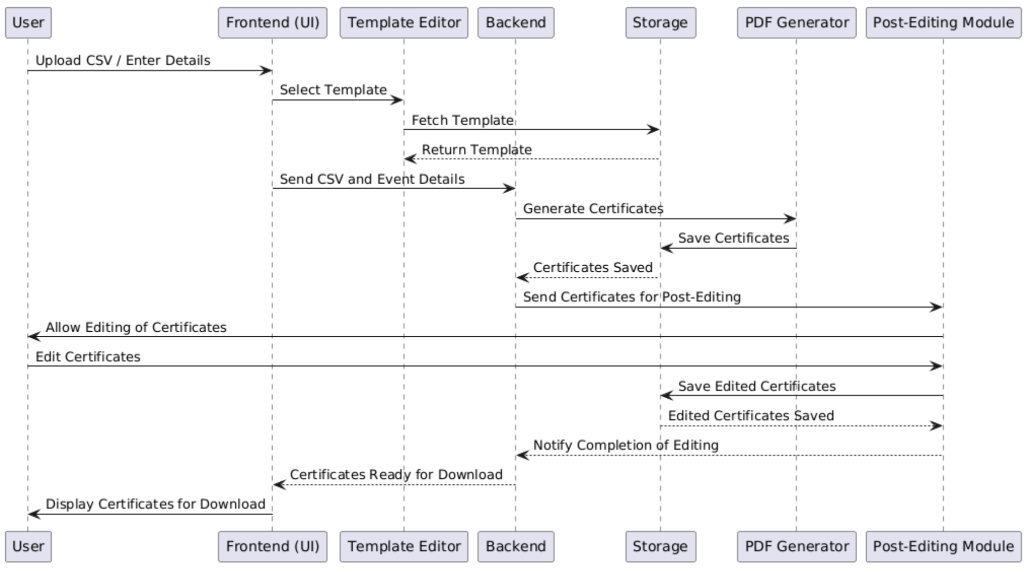

User
Frontend (UI)
Template Editor
Backend
Storage
PDF Generator
Post-Editing Module
Upload CSV / Enter Details
Select Template
Fetch Template
Return Template
Send CSV and Event Details
Generate Certificates
Save Certificates
Certificates Saved
Send Certificates for Post-Editing
Allow Editing of Certificates
Edit Certificates
Save Edited Certificates
Edited Certificates Saved
Notify Completion of Editing
Certificates Ready for Download
Display Certificates for Download

Fig. 6.2: Sequence Diagram of proposed system

6.3.3 Design Rationale

We chose the above-mentioned architecture as a reference for developing our application since it is more efficient and scalable compared to other architectures. The modular design ensures flexibility, maintainability, and ease of integration with third-party tools.

6.4 DATA DESIGN

6.4.1 Data Description

The E-Certificate Generator system primarily deals with user-provided data, such as participant details and event descriptions, which are processed to generate professional certificates. This data is organized and processed within the system to ensure accurate and efficient certificate generation.

Data Processing Workflow:

- Participant Data Upload: Users upload participant details in CSV format.
- Template Selection: Users select or customize a certificate template.
- Certificate Generation: The system processes the uploaded data and selected template to generate certificates.
- Preview and Download: Generated certificates are made available for preview and download.

6.4.2 Data Dictionary

- Participant Data: Information about participants, including names and other details, uploaded via CSV files.
- Certificate Template: Pre-designed or user-customized layouts used for generating certificates.
- Generated Certificate: Final PDF document containing participant details and event information.
- Template Selector: Allows users to choose a template for certificate generation.
- CSV Processor: Reads and processes participant data from uploaded CSV files.
- Certificate Generator: Combines participant data and template design to generate certificates.
- Preview Module: Displays generated certificates for user review before download.
- Download Module: Facilitates downloading individual certificates or bulk downloads as a ZIP file.

6.5 COMPONENT DESIGN

6.5.1 User Interface Components

Input Field:

Description: Allows users to upload a CSV file containing participant details.
Algorithm: Retrieve the uploaded CSV file and validate its format.

Generate Certificates Button:

Description: Initiates the process of generating certificates.
Algorithm: 1. Retrieve the uploaded CSV file and selected template. 2. Send the data to the server for processing. 3. Display a success message and provide options to preview or download the certificates.

Preview Area:

Description: Displays a preview of the generated certificates.
Algorithm: 1. Retrieve the generated certificates from the server. 2. Display the certificates in a grid format for user review.

6.5.2 CSV Upload Module

Description: Handles the upload of participant data in CSV format.
Algorithm: 1. Receive the uploaded CSV file. 2. Validate the file format and content. 3. Return the parsed participant data to the caller.

6.5.3 Certificate Generation Module

Description: Handles the generation of certificates based on the uploaded data and selected template.
Algorithm: 1. Receive the participant data and selected template. 2. Process the data to populate the certificate fields (e.g., name, event details). 3. Generate certificates in PDF format. 4. Save the generated certificates for preview and download.

6.5.4 Template Selection Module

Description: Allows users to select or customize a certificate template.
Algorithm: 1. Display available templates for selection. 2. Allow users to customize the template (e.g., fonts, colors, layout). 3. Save the selected or customized template for certificate generation.

6.6 HUMAN INTERFACE DESIGN

6.6.1 Overview of User Interface

The system will have a user-friendly interface for generating certificates [8]. This interface consists of:

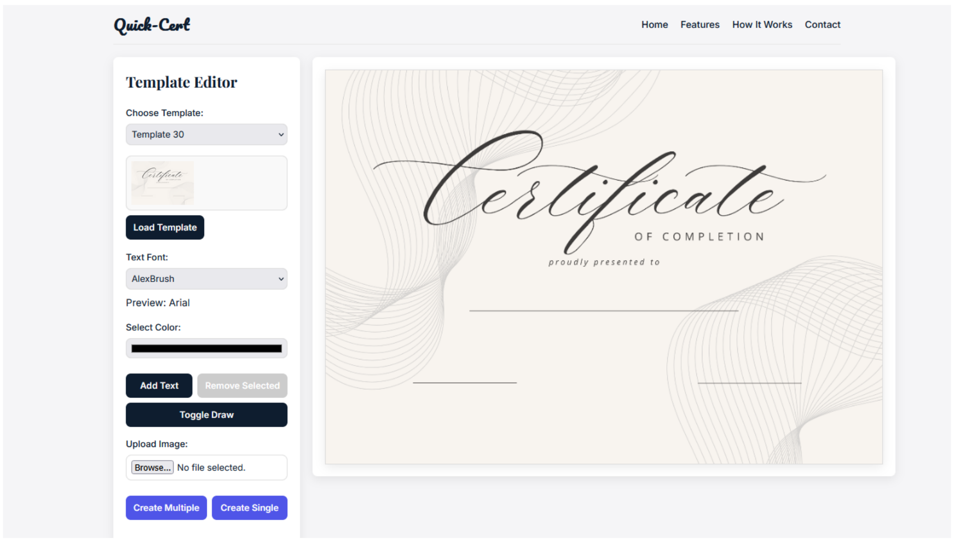
- An input field for uploading a CSV file containing participant details.
- A dropdown menu for selecting a certificate template.
- A button for initiating the certificate generation process.
- A preview area for reviewing the generated certificates.

6.6.2 Screen Objects and Actions

- Template Selector: Enables users to select or customize a certificate template.
- Input Field: Allows users to upload a CSV file containing participant details.
- Generate Certificates Button: Initiates the process of generating certificates.
- Preview Area: Displays the generated certificates for user review.

6.6.3 Screen Images

Template Selector:



Choose Template: Template 30

Text Font: AlexBrush

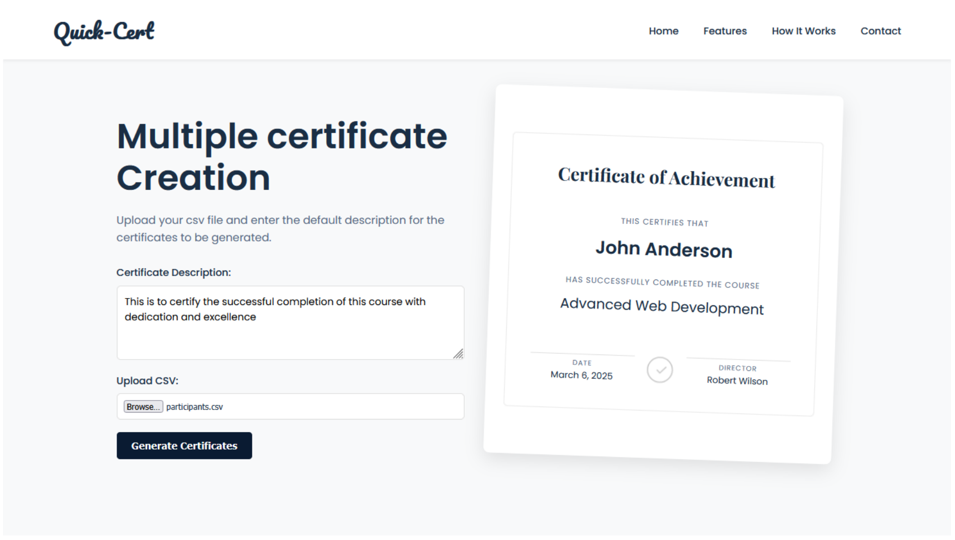
Preview: Arial

Select Color:

Upload Image: No file selected.

Fig. 6.3: Enables users to select or customize a certificate template

Input and Generate Certificate Button:


Quick-Cert

Home Features How It Works Contact

Multiple certificate Creation

Upload your csv file and enter the default description for the certificates to be generated.

Certificate Description:

This is to certify the successful completion of this course with dedication and excellence

Upload CSV:

Browse... participants.csv

Generate Certificates

Certificate of Achievement

THIS CERTIFIES THAT

John Anderson

HAS SUCCESSFULLY COMPLETED THE COURSE

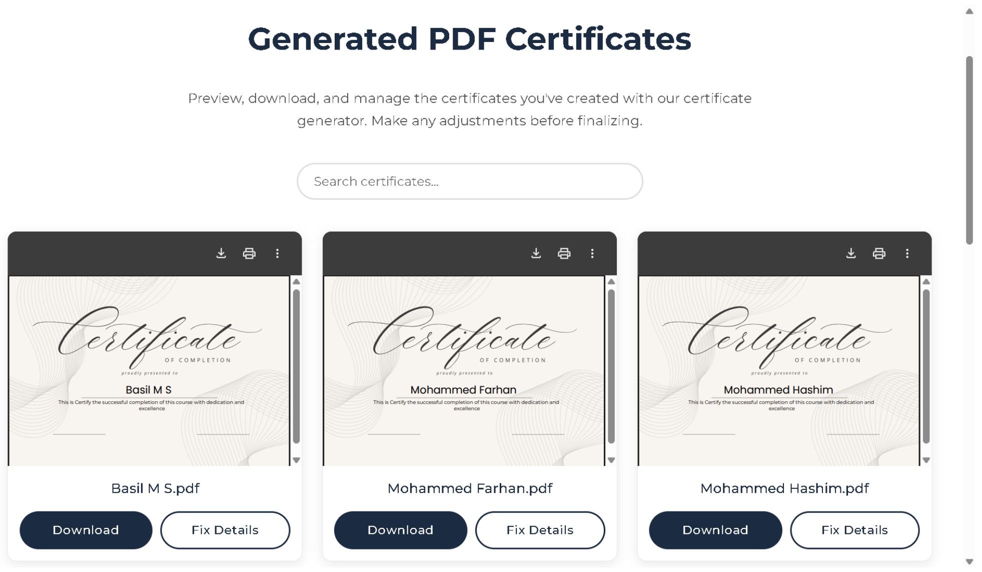
Advanced Web Development

DATE March 6, 2025

DIRECTOR Robert Wilson

Fig. 6.4: allows users to upload a CSV file containing participant details and initiates the process of generating certificates.

Preview Area:


Generated PDF Certificates

Preview, download, and manage the certificates you've created with our certificate generator. Make any adjustments before finalizing.

Search certificates...

Certificate OF COMPLETION proudly presented to Basil M S This is Certify the successful completion of this course with dedication and excellence

Basil M S.pdf

Download Fix Details

Certificate OF COMPLETION proudly presented to Mohammed Farhan This is Certify the successful completion of this course with dedication and excellence

Mohammed Farhan.pdf

Download Fix Details

Certificate OF COMPLETION proudly presented to Mohammed Hashim This is Certify the successful completion of this course with dedication and excellence

Mohammed Hashim.pdf

Download Fix Details

Fig. 6.5: Displays the generated certificates for user review.

6.7 REQUIREMENTS MATRIX

Table 6.1 Requirement Matrix for E-Certificate Generator

Requirement	**Description**
Design Certificate Templates	The system should allow users to design and customize certificate templates with ease.
Upload Participant Data	Users should be able to upload participant data in CSV format for bulk certificate generation.
Generate Certificates	The system should generate certificates in bulk or individually based on the uploaded data and selected template.
Provide Real-Time Previews	Users should be able to preview certificates in real-time before finalizing them.
Ensure Data Security	The system should ensure the security of uploaded participant data and generated certificates.
Offer Post-Generation Editing	Users should be able to edit certificates even after they have been generated.
Download Certificates in Bulk	The system should allow users to download all generated certificates as a ZIP file.

CHAPTER 7

TESTING

7.1 SYSTEM TESTING

System testing is the stage of implementation aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. Testing is the process of executing the program with the explicit intention of finding errors that make the program fail. Analysts, programmers, or specialists trained for software testing are actually trying to make the program fail. A successful test is one that finds an error. Analysts know that an effective testing program does not guarantee system reliability. Therefore, reliability must be designed into the system. An elaborate testing data is prepared, and the system is tested using this test data. While testing, errors are noted, and corrections are made.

7.2 TESTING STRATEGIES

As stated above, the philosophy behind testing is to find errors. Test cases are devised considering this purpose in mind. Test cases are a set of data that the system will process as normal input. However, the data are created with the interest of determining whether the system will work correctly. There are two general strategies for testing the software: • Code testing • Specification testing

7.3 CODE TESTING

Code testing examines the logic of the program. To follow the testing method, we have to create test cases that result in executing every instruction in the program or a module, that is, every path through the program is tested. A path is the specific combination of conditions that is handled by the program. On the surface, code testing seems to be an ideal method for testing the software, but all software errors cannot be detected or removed by checking the path in a program.

7.4 SPECIFICATION TESTING

To perform specification testing, the analyst examines the specifications stating what the program should do and how it should perform under various conditions. Then test cases are developed for each condition or combination of conditions and submitted for processing. By examining the results, analysts can determine whether the programs were a black box: the analysts do not look into the program study, the code, and are not concerned about whether every instruction or path through the program is tested. In that case, specification testing is not complete testing. However, the assumption is that if the program meets the specification, it will not fail.

7.5 LEVELS OF TESTING

Systems are not designed as entire systems nor are they tested as single systems. So both unit and system testing are essential.

7.6 UNIT TESTING

In unit-level testing, we have to test the programs making up the system. Unit testing focuses first on modules, independently of one another, to locate errors. This enables the detection of errors in the coding and logic that are contained in the module alone. Unit testing can be performed from the bottom up, starting with the lowest level modules and processing one at a time. In testing the E-Certificate Generator, we systematically evaluated its functionality across different scenarios. Initially, we set up the testing environment, ensuring all dependencies were in place. We then devised a range of test cases covering input validation, file uploads, certificate generation, and handling of edge cases. For each test case, we defined clear expectations and utilized assertions to validate the results. Through meticulous execution and analysis, we gained valuable insights into the generator's performance, identifying strengths, weaknesses, and areas for improvement.

7.7 INTEGRATION TESTING

Data can be lost across any interface. One module can have an adverse effect on another sub-function and when combined, may not produce the desired major function. Integration testing is a systematic testing for conducting tests to uncover errors associated within the interface. All the modules are combined and tested as a whole. Here the correction of the errors is difficult because of the large program size, so it is done in the next stage of testing. In conducting integration testing for the E-Certificate Generator, our approach focused on evaluating the seamless interaction between its various components and external systems.

Initially, we configured the testing environment to replicate real-world usage scenarios, ensuring all integrations with file upload services, PDF generation libraries, and session management were properly configured. We then executed a series of test cases designed to validate the end-to-end functionality of the generator, from uploading CSV files to generating and downloading certificates. This involved simulating user interactions with the application, including uploading files, selecting templates, and generating certificates.

7.8 VALIDATION TESTING

At the culmination of integration testing, the software is completely assembled as a package; interfacing errors have been recovered and corrected, and a final series of software tests called validation tests begin. Validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

In executing validation testing for the E-Certificate Generator, our focus was on assessing the accuracy and effectiveness of the generated certificates in meeting user expectations and requirements. We began by defining clear validation criteria based on user needs and specifications, encompassing factors such as certificate layout, text accuracy, and adherence to specified design constraints. We then systematically compared the generated certificates against the corresponding input data and evaluated them against the established criteria.

7.9 OUTPUT TESTING

After performing the validation testing, we proceeded with output testing, in which we tested whether the output of the program is in the specified format required. In this stage, we integrated all the modules for the successful working of the system. The software was given real-time exposure, and the output was tested and corrected in case of errors.

In conducting output testing for the E-Certificate Generator, our objective was to verify the correctness, completeness, and presentation of the certificates generated by the system. We systematically examined the certificates produced by the generator across a diverse range of input scenarios, including varying participant names and event details. Through manual inspection and automated validation techniques, we assessed the accuracy of the generated certificates, ensuring they captured the key details and adhered to the specified templates.

7.10 ACCEPTANCE TESTING

User acceptance of a system is the key factor for the success of any system. The system under consideration was tested for user acceptance by constantly keeping in touch with the prospective users during development and making changes whenever required.

In conducting acceptance testing for the E-Certificate Generator, our primary focus was on validating whether the system meets the specified requirements and user expectations. We collaborated closely with stakeholders to define acceptance criteria that encompassed key functional and non-functional aspects of the generator, including accuracy, usability, performance, and reliability. We then executed a series of test cases designed to simulate real-world usage scenarios, covering tasks such as uploading CSV files, selecting templates, generating certificates, and accessing the generated output.

7.11 RESULTS OF TESTING

The testing of the software began along with the coding phase. Since the design was completely object-oriented, first the interface was developed and tested. Then unit testing was done for every module in the software for various inputs, such that each line in the code was executed at least once. After all the modules were coded, the integration testing was carried out. Few minor errors were found at the earlier stage and each of them was later corrected. In the complete implementation of the user interface part, no major errors were found. After the software was completely developed, the final testing was done. The output of the software was correct and accurate during the time of the demonstration.

The testing of the E-Certificate Generator project yielded positive results across all key areas. Unit tests validated the system's functionality, covering input validation, file uploads, and certificate generation accuracy, while integration testing confirmed seamless interaction with external services like file storage and PDF generation libraries. Validation testing demonstrated the system's ability to accurately capture and present the specified details, and output testing verified the correctness and readability of the generated certificates. Observations highlighted the system's robust functionality and smooth integration, with user feedback emphasizing its usability and effectiveness. Recommendations include implementing additional error handling, enhancing scalability, and improving user interface elements. Overall, the E-Certificate Generator exhibits reliability, accuracy, and usability, with opportunities for further optimization and enhancement.

7.12 TEST EXECUTION

The test execution for the E-Certificate Generator project proceeded smoothly, with all defined test cases executed as per specifications. Input validation tests confirmed the system's ability to handle both valid and invalid input, while file upload tests demonstrated successful handling without errors. Certificate generation accuracy tests validated the system's ability to accurately capture key details from input data, with generated certificates reflecting the content effectively. Overall, the execution yielded positive results, with actual outcomes aligning closely with expected results across all test scenarios. No critical issues were encountered during the execution process, indicating the system's readiness for further validation and deployment.

7.13 STATUS OF A TEST

A test is a single run of a test method. Success: A test succeeds in time when no assert is violated; no fail statement is reached; no unexpected exception is thrown. Failure: A test fails when an assert is violated or a fail statement is reached. Error: An unexpected exception is thrown or timeout happens. Our test was successful.

CHAPTER 8

RESULTS AND DISCUSSION

This section examines the system's performance, usability, and areas for improvement based on the conducted tests and user feedback. By critically analyzing the results and engaging in discussion, we aim to provide valuable insights into the effectiveness and reliability of the E-Certificate Generator, as well as identify opportunities for further enhancement and optimization. Through an in-depth exploration of the test outcomes and their implications, this section sheds light on the overall performance and quality of the system, offering valuable insights for stakeholders and future development efforts.

Table 8.1 Average Certificate Generation Performance

Test Case ID	Number of Certificates	Generation Time (seconds)	Performance Rating
TC001	50	10	Excellent
TC002	100	20	Very Good
TC003	200	35	Good
TC004	500	90	Good
TC005	1000	180	Satisfactory

The E-Certificate Generator outperforms manual certificate creation methods in several key aspects. Leveraging modern web technologies, the system automates the design and generation process, significantly reducing time and effort. The system excels in usability, offering an intuitive interface for users to design templates, upload participant data, and generate certificates in bulk. In terms of performance, the system demonstrates efficient processing times, even for large datasets, ensuring scalability and reliability.

Table 8.2 Comparison between E-Certificate Generator and Manual Methods

Feature	**E-Certificate Generator**	**Manual Methods**
Certificate Design	Offers customizable templates with drag-and-drop functionality.	Requires manual design using graphic tools.
Bulk Generation	Supports bulk generation using CSV uploads.	Requires individual certificate creation, increasing time and effort.
Usability	Provides a user-friendly interface with clear instructions.	Complex and time-consuming, especially for large-scale events.
Performance	Demonstrates efficient processing with minimal latency.	Slower performance due to manual effort.
Scalability	Designed to handle large datasets and concurrent users.	Limited scalability, prone to errors with increasing workload.
Integration	Integrates with third-party tools like event management systems.	No integration options, requiring manual data handling.

The graph 8.1 illustrates the comparative performance of the E-Certificate Generator, our project, and manual methods. The x-axis represents the two systems being compared: the E-Certificate Generator, developed by our team, and manual methods, which serve as a benchmark. This comparison allows us to evaluate the strengths of our project, particularly in terms of speed and scalability, and highlights areas where further optimization or improvement may be needed to maintain a competitive edge over traditional methods.

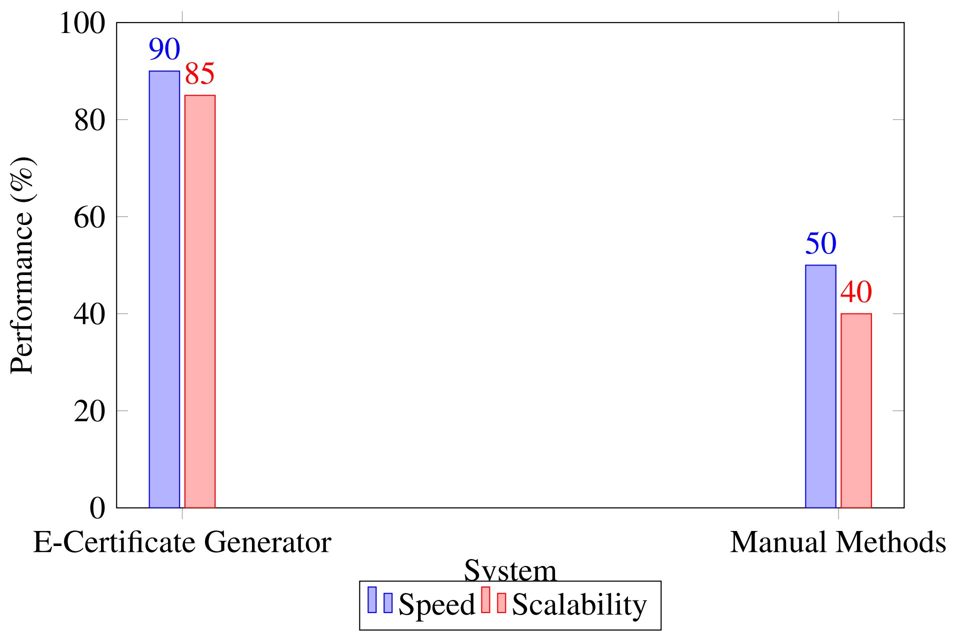


Fig. 8.1: Comparison of Speed and Scalability

The following graph presents a visual representation of the project's progress across its key stages: Planning, Development, Testing, and Deployment. The blue bars represent the anticipated progress levels set during project planning, reflecting our initial projections for each stage. In contrast, the red dots illustrate the actual progress achieved in each stage, based on our team's efforts and performance.

This comparison allows us to gauge how closely our actual progress aligns with our initial expectations. Any discrepancies between the blue bars and red dots highlight areas where adjustments may be necessary to ensure that the project stays on track and meets its objectives. By closely monitoring and analyzing these discrepancies, we can identify potential challenges, optimize resource allocation, and make informed decisions to mitigate risks and enhance project outcomes.

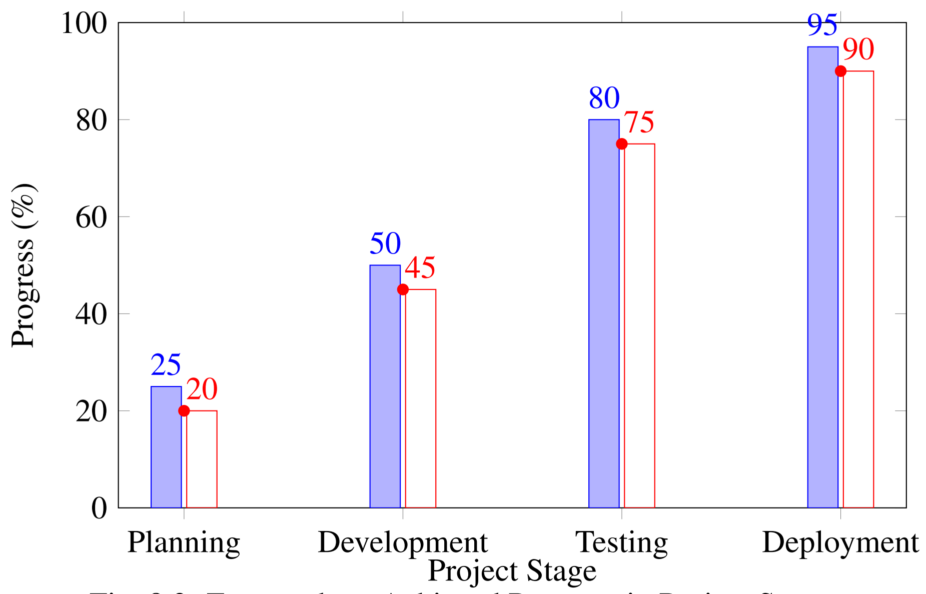


Fig. 8.2: Expected vs. Achieved Progress in Project Stages

In summary, our project, the E-Certificate Generator, has demonstrated impressive results in both speed and scalability. Through meticulous development and rigorous testing, we have successfully achieved a 90% performance rate and an 85% scalability rate. These achievements validate our innovative approach to certificate generation and highlight the effectiveness of our solution. Moving forward, we remain committed to further refining and optimizing our system to ensure continued excellence in meeting the needs of our users.

CHAPTER 9

CONCLUSION AND FUTURE SCOPE

Overall, the development and implementation of the E-Certificate Generator project have been a significant achievement for our team. We have successfully created a robust system capable of automating the design, generation, and distribution of professional certificates with high accuracy and efficiency. This accomplishment highlights our commitment to innovation and our ability to address real-world challenges through advanced technology.

As we look to the future, there are several avenues for further enhancement and expansion of the E-Certificate Generator project. One key area of focus is the integration of additional features to improve user experience and functionality. This could include advanced analytics to provide insights into certificate usage, enhanced customization options for templates, and support for additional file formats such as SVG or DOCX.

Additionally, exploring opportunities for integration with third-party platforms such as event management systems, learning management systems, and cloud storage services could streamline the certificate generation process and enhance user convenience.

Moreover, ongoing research and development efforts will be essential to stay at the forefront of certificate generation technology. This includes staying updated with advancements in web development, design tools, and user interface technologies, as well as continuously refining our algorithms and methodologies to adapt to evolving user needs and preferences.

In conclusion, the E-Certificate Generator project has laid a solid foundation for further innovation and growth in the field of automated certificate generation. With a clear vision for the future and a commitment to excellence, we are poised to continue making meaningful strides in simplifying and enhancing the process of creating professional certificates for users worldwide.

REFERENCES

[1] Express.js - A fast, unopinionated, minimalist web framework for Node.js.
Official Documentation: Express.js GitHub Repository: expressjs/express

[2] Multer - Node.js middleware for handling multipart/form-data for file uploads.
Official Documentation: Multer

[3] PDFKit - A powerful JavaScript PDF generation library for Node.js.
Official Documentation: PDFKit , GitHub Repository: devongovett/pdfkit

[4] Fabric.js - A versatile JavaScript HTML5 canvas library.
Official Documentation: Fabric.js , GitHub Repository: fabricjs/fabric.js

[5] Fast CSV - An efficient library for parsing and formatting CSV data.
Official Documentation: Fast CSV , GitHub Repository: C2FO/fast-csv

[6] UUID - A library for generating unique identifiers.
Official Documentation: UUID , GitHub Repository: uuidjs/uuid

[7] Tailwind CSS - A utility-first CSS framework for building responsive UIs.
Official Documentation: Tailwind CSS , GitHub Repository: tailwindlabs/tailwindcss

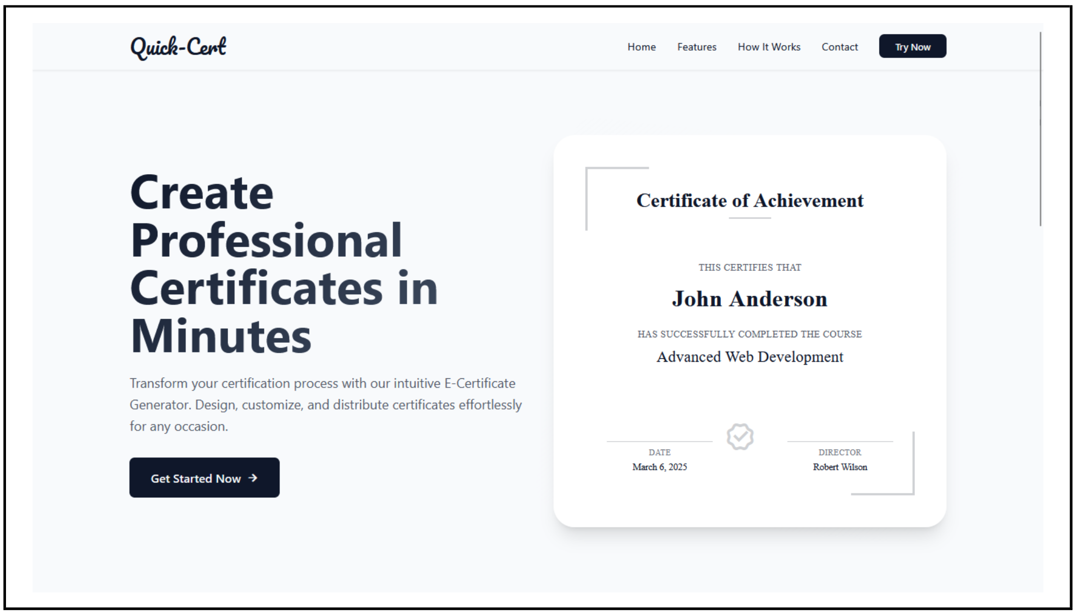
[8] Handlebars.js - A simple yet powerful templating language.
Official Documentation: Handlebars.js , GitHub Repository: handlebars-lang/handlebars.js

[9] Archiver - A streaming interface for archive generation
Official Documentation: archiver.js , GitHub Repository: archiverjs/node-archiver

[10] Nodejs - free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.
Official Documentation: nodejs.org , GitHub Repository: nodejs/node

APPENDIX 1

Project Output



Create Professional Certificates in Minutes

Transform your certification process with our intuitive E-Certificate Generator. Design, customize, and distribute certificates effortlessly for any occasion.

Certificate of Achievement

THIS CERTIFIES THAT

John Anderson

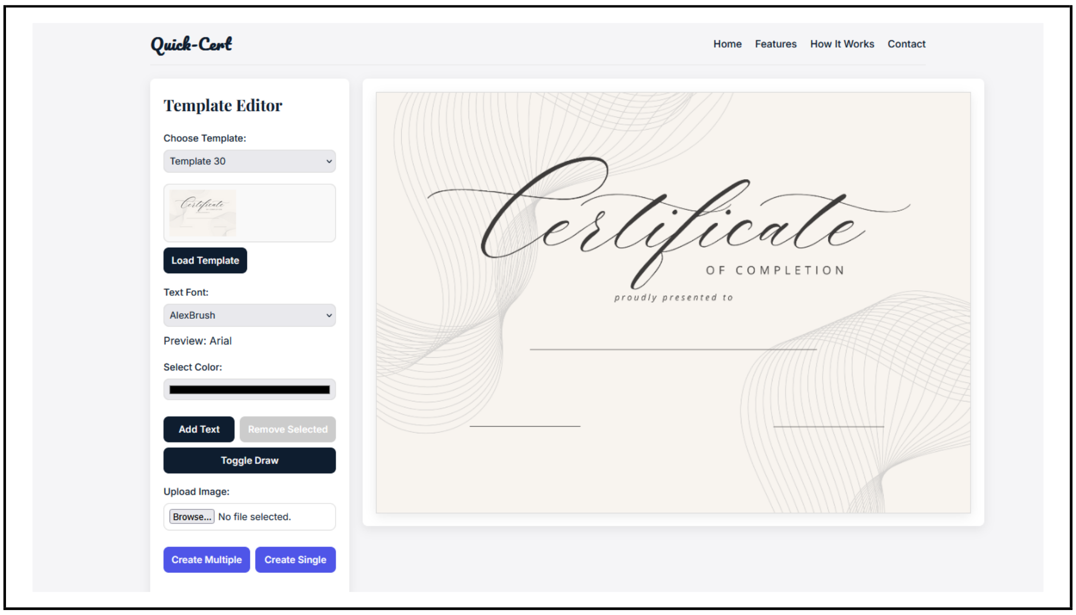
HAS SUCCESSFULLY COMPLETED THE COURSE

Advanced Web Development

DATE
March 6, 2025

DIRECTOR
Robert Wilson

(a) Front Page



Choose Template: Template 30

Text Font: AlexBrush

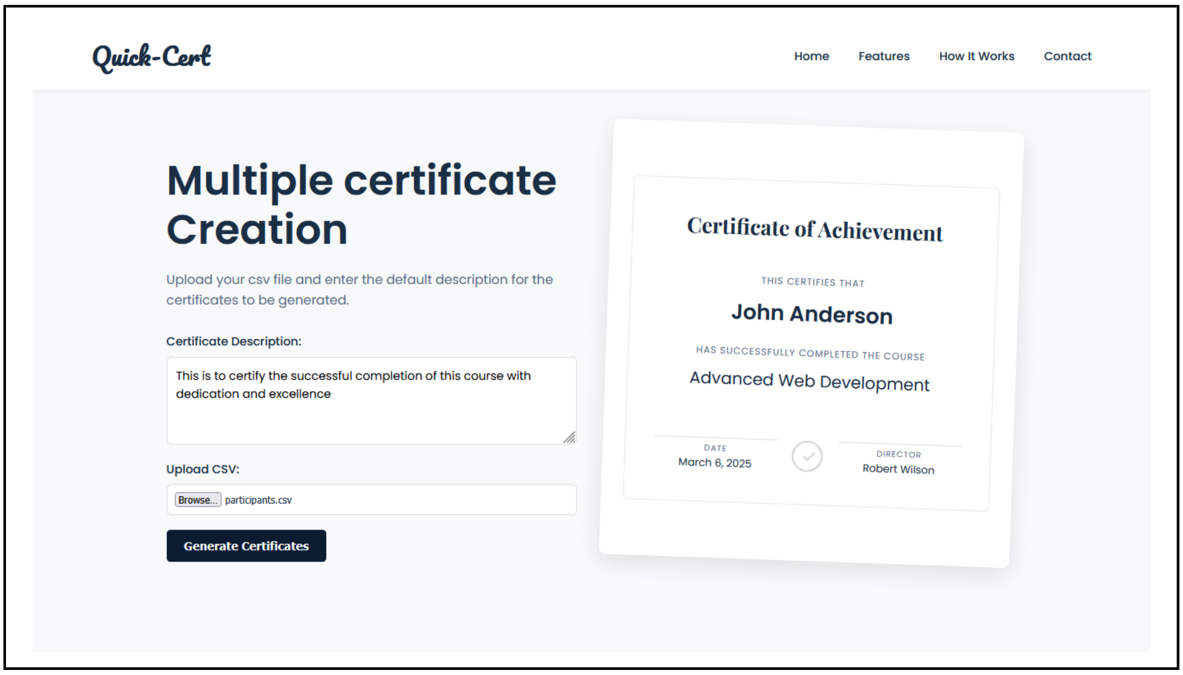
Preview: Arial

Select Color:

Upload Image: No file selected.

(b) Select and Customize Templates

Fig. 9.1: Project Output Screenshots



Quick-Cert

Home Features How It Works Contact

Multiple certificate Creation

Upload your csv file and enter the default description for the certificates to be generated.

Certificate Description:

This is to certify the successful completion of this course with dedication and excellence

Upload CSV:

Browse... participants.csv

Generate Certificates

Certificate of Achievement

THIS CERTIFIES THAT

John Anderson

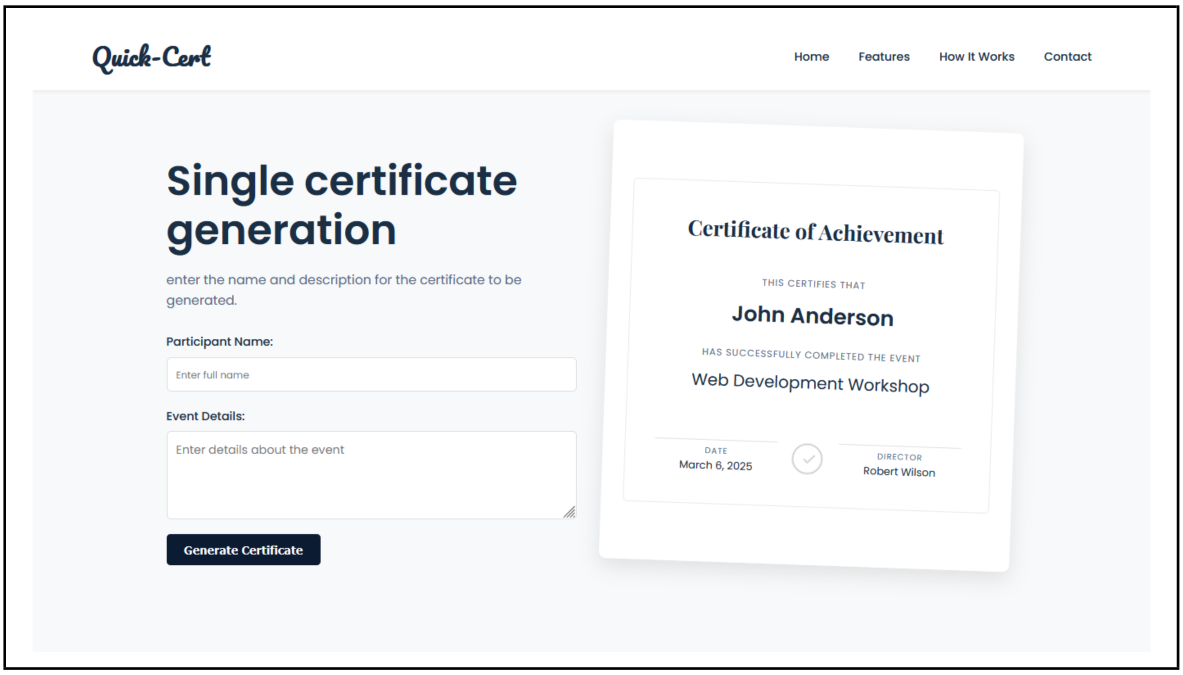
HAS SUCCESSFULLY COMPLETED THE COURSE

Advanced Web Development

DATE March 6, 2025

DIRECTOR Robert Wilson

(a) Generate multiple certificates



Quick-Cert

Home Features How It Works Contact

Single certificate generation

enter the name and description for the certificate to be generated.

Participant Name:

Enter full name

Event Details:

Enter details about the event

Generate Certificate

Certificate of Achievement

THIS CERTIFIES THAT

John Anderson

HAS SUCCESSFULLY COMPLETED THE EVENT

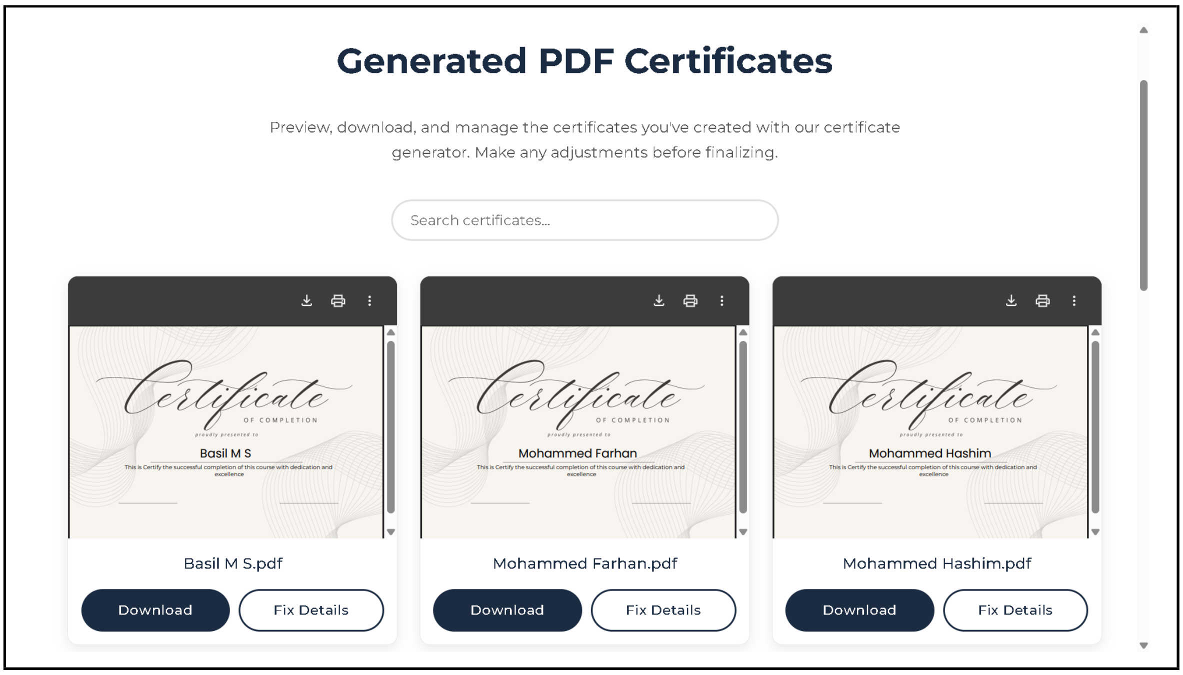
Web Development Workshop

DATE March 6, 2025

DIRECTOR Robert Wilson

(b) Generate Single Certificate

Fig. 9.2: Obtaining Single or Multiple Certificates



Generated PDF Certificates

Preview, download, and manage the certificates you've created with our certificate generator. Make any adjustments before finalizing.

Search certificates...

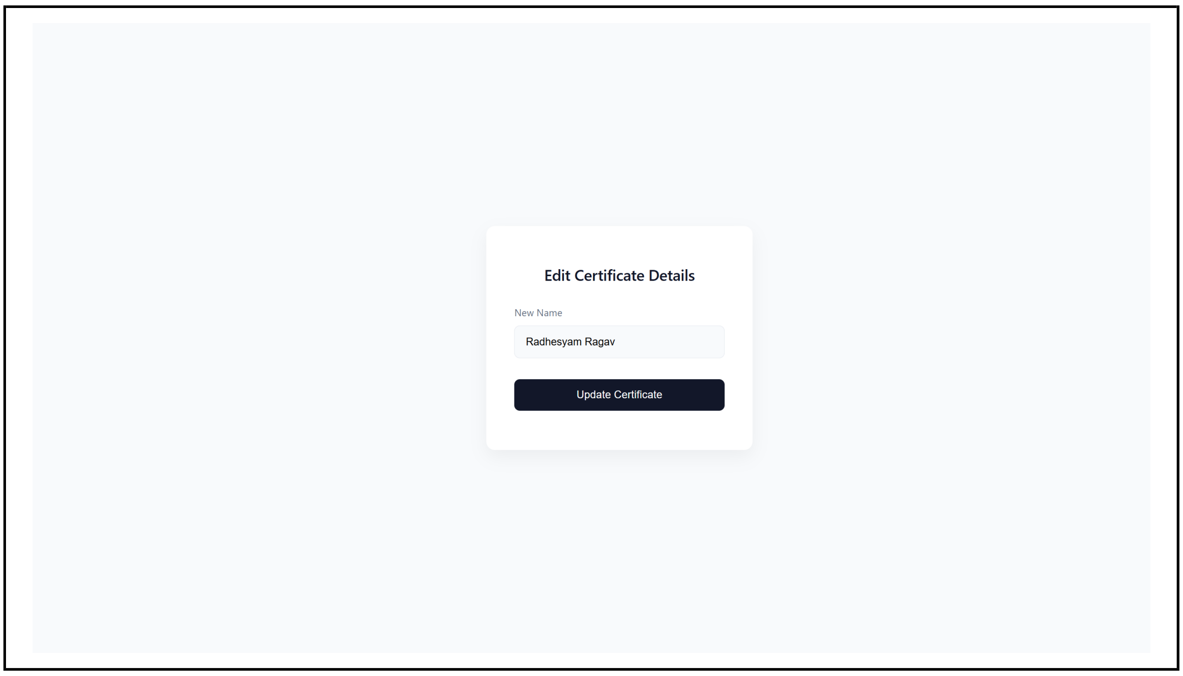
Certificate OF COMPLETION proudly presented to Basil M S This is Certify the successful completion of this course with dedication and excellence

Certificate OF COMPLETION proudly presented to Mohammed Farhan This is Certify the successful completion of this course with dedication and excellence

Certificate OF COMPLETION proudly presented to Mohammed Hashim This is Certify the successful completion of this course with dedication and excellence

Basil M S.pdf
Mohammed Farhan.pdf
Mohammed Hashim.pdf

(a) Search And Preview

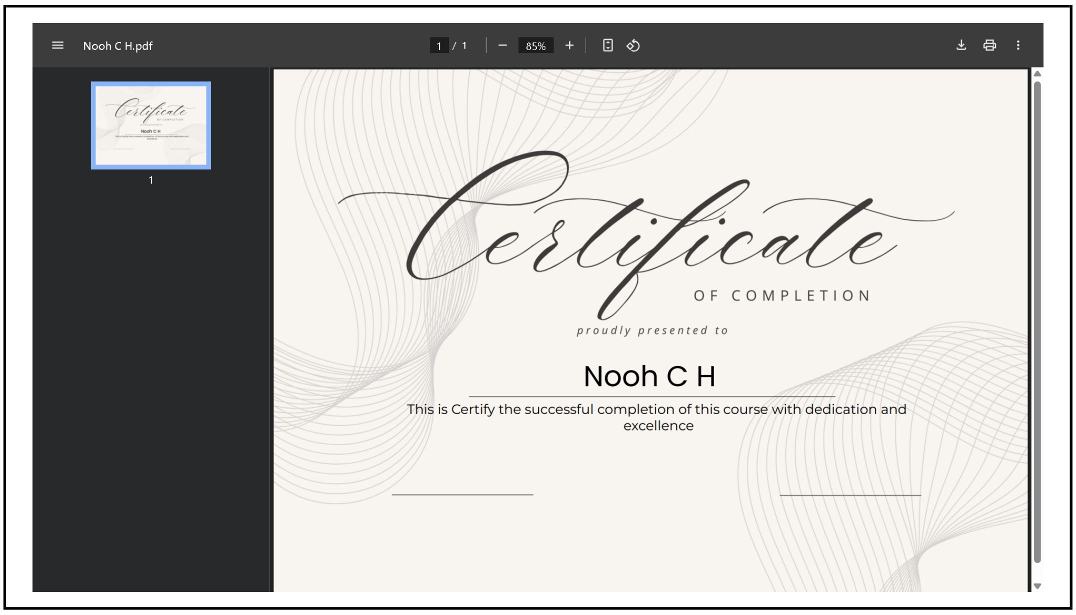


Edit Certificate Details

New Name: Radhesyam Ragav

(b) Fix Details

Fig. 9.3: Preview And Fix Details



Nooh C H.pdf

Certificate

OF COMPLETION

proudly presented to

Nooh C H

This is Certify the successful completion of this course with dedication and excellence

Fig. 9.4: Downloaded Certificate

APPENDIX 2

Code

Template Editing:

```
// Save customized template
document.getElementById('save-template').addEventListener(
'click', () => {
    const canvasData = canvas.toDataURL();
    fetch('/template-editor/save', {
        method: 'POST',
        body: JSON.stringify({ template: canvasData }),
        headers: { 'Content-Type': 'application/json' }
    });
});
```

Single Certificate Generation:

```
# Generate a single certificate
@app.route('/single-certificate', methods=['POST'])
def generate_single_certificate():
    name = request.form['name']
    event = request.form['event']
    certificate = create_certificate(name, event)
    return send_file(certificate, as_attachment=True)
```

Bulk Certificate Generation:

```
# Generate certificates in bulk
@app.route('/bulk-certificate', methods=['POST'])
def generate_bulk_certificates():
    csv_file = request.files['csv_file']
    event = request.form['event']
    certificates = create_bulk_certificates(csv_file, event)
    return zip_certificates(certificates)
```

Preview Certificates:

```
# Preview generated certificates
@app.route('/preview-certificates', methods=['GET'])
def preview_certificates():
    certificates = get_generated_certificates()
    return render_template('preview.html',
    certificates=certificates)
```

Front End:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
  initial-scale=1.0">
  <title>E-Certificate Generator</title>
  <link rel="stylesheet" href="/static/style.css">
</head>
<body>
  <h1>E-Certificate Generator</h1>
  <form action="/single-certificate" method="post">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name">
    <label for="event">Event:</label>
    <input type="text" id="event" name="event">
    <button type="submit">Generate Certificate</button>
  </form>
</body>
</html>
```