Minor Project

on

Applica Auto

submitted in partial fulfillment of the requirements
for the award of the degree
of

Bachelor of Technology

in

Computer Science and Engineering

By

Adarsh Sharma
Enrollment No. A60205220107

Gaurav Panwar
Enrollment No. A60205220105

Under the guidance of

Dr. Dinesh Sharma
Assistant Professor

Department of Computer Science and Engineering
Amity School of Engineering & Technology
Amity University Madhya Pradesh, Gwalior
June 2023

Department of Computer Science and Engineering
Amity School of Engineering and Technology
Amity University Madhya Pradesh, Gwalior

DECLARATION

We, **Adarsh Sharma and Gaurav Panwar**, students of Bachelor of Technology in Computer Science and Engineering hereby declare that the Project report entitled **"Applica Auto"** which is submitted by us to Computer Science and Engineering, Amity School of Engineering & Technology, Amity University Madhya Pradesh, in partial fulfillment of the requirement for the award of the Degree of Bachelor of Technology in Computer Science and Engineering, has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition. Our supervisor, HOD, and the Institute should not be held for full or partial violation of copyrights if found at any stage of our degree.

Adarsh Sharma

Date:30May,2023

(Enrollment No: A60205220107)

Gaurav Panwar

(Enrollment No: A60205220105)

Department of Computer Science and Engineering
Amity School of Engineering and Technology
Amity University Madhya Pradesh, Gwalior

CERTIFICATE

This is to certify that the major project entitled "**Applica Auto**" by **Adarsh Sharma (Enrollment No A60205220107) and Gaurav Panwar(Enrollment No A60205220105)** is a bonafide record of the project carried out by them under my supervision and guidance in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering in the Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Madhya Pradesh, Gwalior. Neither this project nor any part of it has been submitted for any degree or academic award elsewhere.

Date:31 May,2023

Dr. Dinesh Sharma
Assistant Professor

ABSTRACT

Our country's literacy rate is increasing day by day, more than half of the population don't know how to write an application by their own. Instead of enough knowledge some people are not able to write due to lack of confidence and fear of being insulted which raises a very huge problem in official curriculum, as it is impossible for organizations to process without applications because it is a key to communicate officially for any demands or relieves. This problem causes dependencies of the vast group of people either to some selected people who can write the correct application or to the INTERNET.

We aim to develop a web application which basically generates an application automatically just by knowing the basic and personal information at once which will be stored in database already. There are different types of application available according to the choices of the concerned authorities which will help one to easily access the applications at a single platform. Also, applications will be available in different languages keeping different regional people in mind like, Gujarati, Marathi, Telugu, Tamil etc...

In this App, the users generate a grammatically correct application by their own which can be sent to concerned authorities and they get notify if accepted or rejected. In easy words, we can say the users get all the status about their application. Status will make the app efficient with time because the fact is applications should be strict with the time always. One of the most important things we kept in mind is availability of applications at no cost unlike other platforms where they cost a lot for accessing their applications and our app provides security of the personal information which user shared. This makes our app trustworthy and not fragile.

Key words: INTERNET, Application, Web App, Server, Database.

LIST OF FIGURES

LIST OF ABBREVIATIONS

S. No.	Terms	Expanded Form
1	HTML	HyperText Markup Language
2	CSS	Cascading Style Sheets
3	JS	JavaScript
4	MERN	MongoDb, Express js, React JS, Node js
5	DB	DataBase
6	HTTP	Hypertext Transfer Protocol
7	WWW	World Wide Web
8	API	Application Programming Interface
9	IP	Internet Protocol
10	JSON	Javascript Object Notation
11	TCP	Transfer Control Protocol
12	GUI	Grahpical User Interface
13	DOM	Document Object Model

CONTENTS

Chapter 1

Introduction

The web-based Technology named Application autometer, is unified for the use for the use of all and facilitates students and the parents of the students. The system allows the user to write the complex application and letters to the officials by just asking a handful of questions and will get the verification from the officials more efficiently and in all this process user (Applicant) will also have the complete information regarding the verification of the application in the officials. This project is a prototype for all the people who wanted to write the application of any level with ease and get their response effectively and reduce paperwork.

Figure 1.1 Automation of an Application

Perspective of this project:

1. From student/Parents Point of View:
   i. Upcoming Events remainders: Students/Parents will be able to check the upcoming event and the last date for applying in the event in which the application and verification form the parents are required for participation of the students.
   ii. User profile: The user id and password will be provided by the officials to maintain the database of every individual in the system.
   iii. Unified platform: The user will be able to check all the progress and will get notified as soon as their application gets approval on any device which can run thewebsite.

2. From Admin Point of View:

Admin would be able to approve the application sent by the student or by the parent after the approval of the admin the notification will be sent to the student as well as the application will be sent to the higher authority.

3. From Head/Principal Point of View:

The higher authority plays the important role in our system after the approval of the admin the application reaches the higher authority on our portal in which the authority will study the application and according to their choice will be able to accept or deny the application and as per the result, a notification will be sent to admin as well as the student account.

Additionally, for operating the Auto Applica project is a web-based application developed to cater to the needs of college and school students in generating various types of applications. It aims to simplify and automate the application generation process, making it convenient and timesaving for students. By utilizing modern web technologies such as React.js, Tailwind CSS, Node.js, Express.js, and MongoDB, the project offers a comprehensive solution to streamline the application generation workflow.

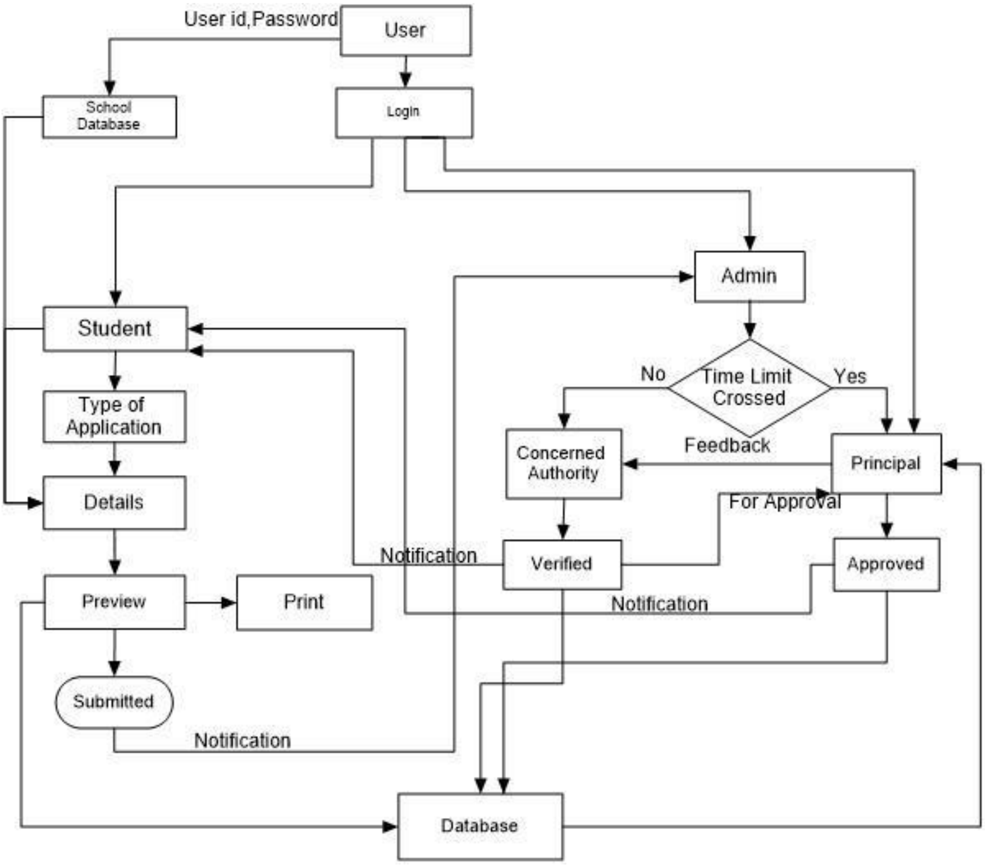


Figure 1.2 Dataflow Diagram

CHAPTER 2

LITERATURE REVIEW

Automation of learning and education are much in demand especially after COVID as there is no scope of offline learning and documentations related stuff. In 1999, E-learning concept was introduced by an educator and researcher Elliot Masie which evokes how people use technology to learn, create and improve the education system by which the idea of generation of automated applications and documentations comes in race. According to the recent studies and research by [1], two major questions were focused, how will a generation of natively digital students react to the rise and spread of these technologies? and how can students use automation to optimize their time and focus on accomplishing their academic goals? Which gives this project a clear picture.

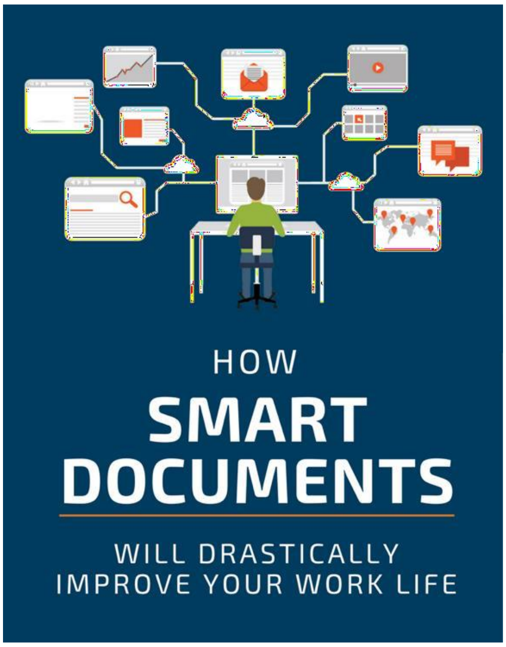


Figure 2.1 Smart Documents

According to Alcatel Lucent [2], "digital transformation is a physical and philosophical change designed to meet the ever-growing demands of the students, faculty, and campus to create a learning environment where everything connects. This is an ecosystem that combines technology, services, and security to bridge the digital gap to create collaborative, interactive and personalized learning experiences". This signifies the importance of digital transformation which is basic motive of the project.

Also, it is essential to understand the importance of Smart Education which is better explained in [3] which also gives the fact that textbooks are eventually replaced by smart devices and new technologies. These all studies evoke the fact that it's the time for improvising the method of teaching and its done through minimizing the paper works done on regular basis like applications for leaves, TC, migration etc. either by student or teacher. It is the part of tool that automates the working of schools or universities which is described in [4] appropriately.

The Myth across human mind is Automation is expensive, but the truth is Investing in automation is like purchasing profit for the long term. As the matter of fact, reduced documentation in offline mode leads to decrease in amount of paper and time. This is pure example of "less input with maximal and effective output."

Figure 2.2 E-Learning

CHAPTER 3

MATERIAL AND METHODS

To achieve the fully functional web application, ReactJs is used for providing the structure to it and TailWind CSS is applied for making the web application interactive as well as attractive. Furthermore, to make flawless communication between frontend and backend, NodeJs is used for managing the backend as ReactJs is most compatible with NodeJs out of any other languages.

Figure 3.1 Frontend – Backend

A powerful combination for managing data in student and teacher applications would be React.js, Node.js, Tailwind CSS, and MongoDB. React.js allows for the creation of dynamic user interfaces, while Node.js serves as the backend server for handling data operations. Tailwind CSS provides a utility-first CSS framework for designing visually appealing interfaces, and MongoDB offers a scalable and efficient NoSQL database solution. Together, these technologies enable the development of modern, efficient, and scalable web applications that can handle data management, create interactive UIs, and handle backend operations seamlessly.

3.1. MATERIAL AND METHOD IN FRONTEND

Technologies used in frontend part are React Js (React Java Script) and Tailwind CSS (Cascading style sheets)

The front-end implementation of Auto Applica is carried out using React.js and CSS Tailwind. React components are developed to create the user interface, handle user interactions, and leverage the styling capabilities of CSS Tailwind. The implementation involves the following key aspects:

3.1.1 Component Development:

React components are created to represent different parts of the application, such as the application form, template selection, and generated application display. Each component encapsulates its own logic and rendering, allowing for modularity and reusability. Components are designed using JSX syntax and are styled using CSS Tailwind classes.

3.1.2 User Interface Design:

CSS Tailwind is used extensively for styling the user interface components. Tailwind provides a utility-first approach to styling, allowing developers to apply pre-defined utility classes to HTML elements. These classes enable quick and efficient styling without the need for writing custom CSS stylesheets. Tailwind's utility classes cover a wide range of styling options, including colors, typography, spacing, and responsive design.

3.1.3 Form Validation:

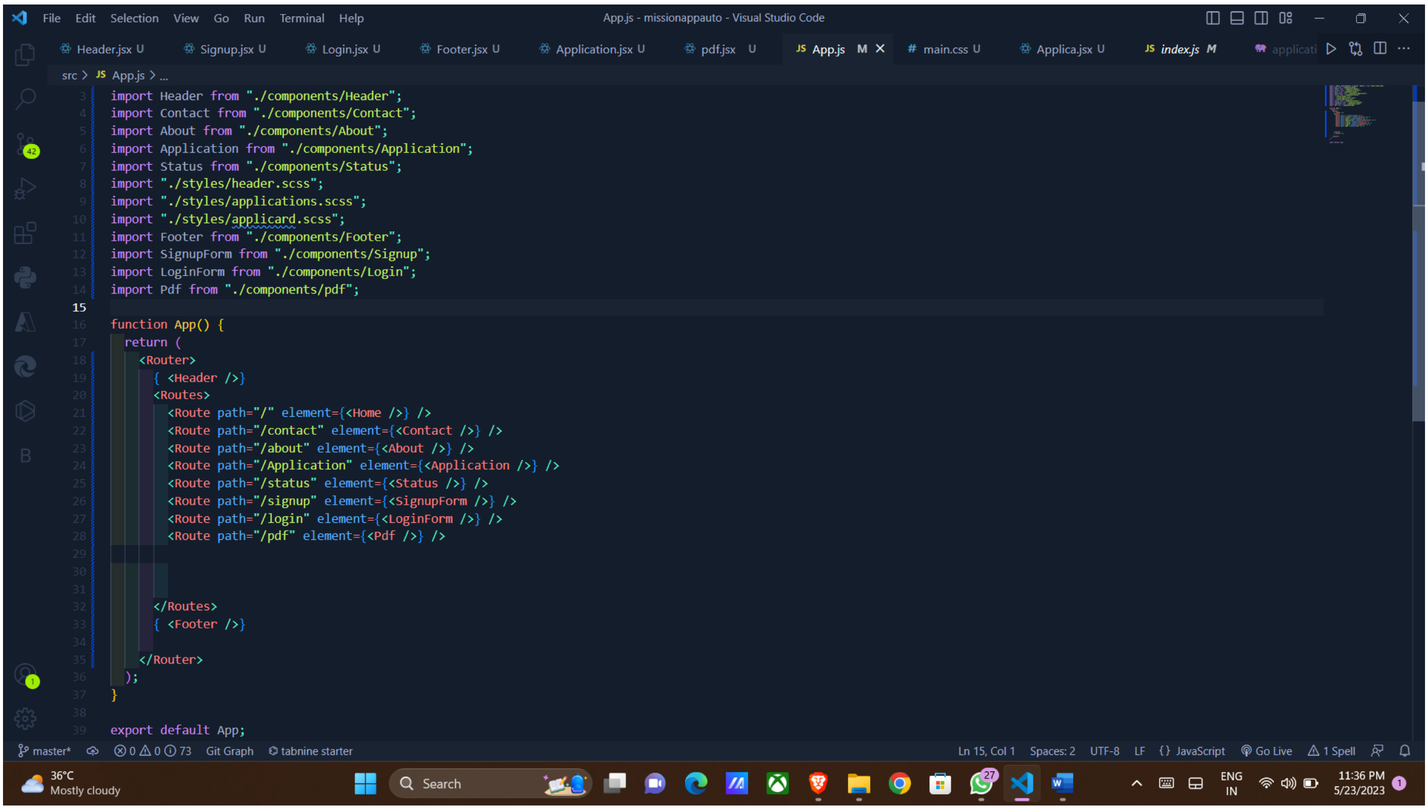
Input validation is implemented on the frontend using React.js and CSS Tailwind classes. Tailwind's form validation classes can be applied to input fields to provide visual feedback to users based on the validity of their input. For example, classes like `valid` and `invalid` can be applied dynamically based on the validation status of the form fields, ensuring that users are aware of any errors or invalid input.

3.1.4 API Integration:

The frontend communicates with the backend APIs to send and receive data using React.js and CSS Tailwind. API integration involves making HTTP requests to the appropriate endpoints and handling responses. CSS Tailwind classes can be applied dynamically to show loading states or success/error messages, providing visual feedback to users during API calls.

3.1.5 Responsive Design:

CSS Tailwind offers responsive design utilities, allowing the application to adapt to different screen sizes and devices. By applying Tailwind's responsive classes, components can be styled differently based on the viewport size. This ensures that the Auto Applica application is accessible and usable across various devices, including desktops, tablets, and mobile phones.



```
import Header from "./components/Header";
import Contact from "./components/Contact";
import About from "./components/About";
import Application from "./components/Application";
import Status from "./components/Status";
import "./styles/header.scss";
import "./styles/applications.scss";
import "./styles/applicard.scss";
import Footer from "./components/Footer";
import SignupForm from "./components/Signup";
import LoginForm from "./components/Login";
import Pdf from "./components/pdf";

function App() {
  return (
    <Router>
      { <Header />}
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/contact" element={<Contact />} />
        <Route path="/about" element={<About />} />
        <Route path="/Application" element={<Application />} />
        <Route path="/status" element={<Status />} />
        <Route path="/signup" element={<SignupForm />} />
        <Route path="/login" element={<LoginForm />} />
        <Route path="/pdf" element={<Pdf />} />



      </Routes>
      { <Footer />}

    </Router>
  );
}

export default App;
```

Figure 3.1.1 React Application

3.2 MATERIAL AND METHOD IN BACKEND

Technologies used in Backend part are NodeJs (Node Java Script), JavaScript, and NO SQL (No Structured Query language

The backend implementation of Auto Applica revolves around developing the server-side logic using Node.js, Express.js, and MongoDB, while the frontend leverages React.js and CSS Tailwind for the user interface. The backend handles incoming requests from the frontend, validates the provided data, generates the applications based on the selected template, and stores them in the MongoDB database. APIs are implemented using Express.js routing, ensuring proper endpoint handling and request processing.

3.2.1 Request Handling:

The backend receives HTTP requests from the frontend and routes them to the appropriate handlers based on the defined API endpoints. The request handling logic includes parsing request data, validating inputs, and invoking the necessary functions to generate applications.

3.2.2 Application Generation:

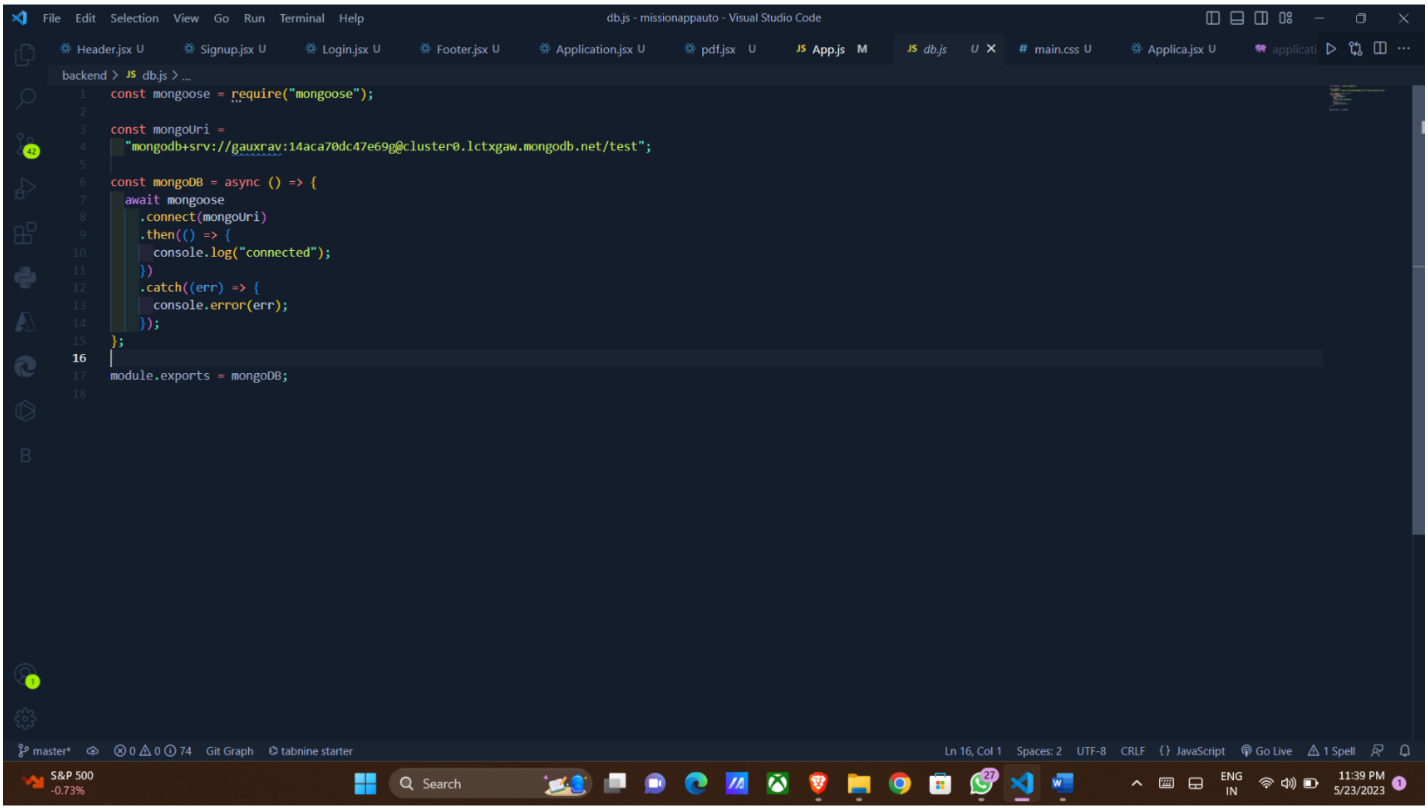
Based on the selected application type and the provided input data, the backend generates the applications. The backend logic formats the data according to the application template, populates the required fields, and creates the final application document. The generation process may involve data manipulation, string concatenation, and conditional logic to handle different application scenarios.

3.2.3 Database Interaction:

Auto Applica utilizes MongoDB as the database for storing and managing the generated applications. The backend interacts with the MongoDB database using a database driver or an Object-Document Mapping (ODM) library like Mongoose. The implementation includes establishing a connection to the database, defining data models, and performing CRUD (Create, Read, Update, Delete) operations on the application records.

3.2.4 Authentication and Authorization

To secure access to the application generation functionality, the backend may implement authentication and authorization mechanisms. This ensures that only authenticated users can generate applications and access their own generated applications. User authentication involves verifying user credentials, managing sessions, and generating access tokens. Authorization mechanisms control the permissions and privileges of users based on their roles and access levels.



File Edit Selection View Go Run Terminal Help
db.js - missionappauto - Visual Studio Code
Header.jsx U
Signup.jsx U
Login.jsx U
Footer.jsx U
Application.jsx U
pdf.jsx U
App.js M
db.js U
main.css U
Applica.jsx U
backend > db.js > ...
master* ⊗ 0 ⚠ 0 ⓘ 74 Git Graph tabnine starter
Ln 16, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live ⚠ 1 Spell
S&P 500 -0.73%
Search
ENG IN
11:39 PM 5/23/2023

```
const mongoose = require("mongoose");

const mongoUri =
  "mongodb+srv://gauxrav:14aca70dc47e69g@cluster0.lctxgaw.mongodb.net/test";

const mongoDB = async () => {
  await mongoose
    .connect(mongoUri)
    .then(() => {
      console.log("connected");
    })
    .catch((err) => {
      console.error(err);
    });
};

module.exports = mongoDB;
```

Figure 3.2.1 Authentication And Authorization

3.3 MATERIAL AND METHOD IN Data Base Management

Auto Applica utilizes MongoDB as the database system for storing the generated applications. The database design and management involve the following steps:

3.3.1 Database Schema Design:

The database schema is designed to define the structure of the application records. It includes the fields required for each application type, such as applicant name, reason for application, start and end dates, and any additional relevant information. The schema ensures consistency and integrity of the data stored in the database.

3.3.2 Data Modelling:

Based on the database schema, data models are created using a library like Mongoose. Data models represent the application records and provide an interface for interacting with the database. They define the data types, validation rules, and relationships between different entities, if applicable.

3.3.3 Database Connection:

The backend establishes a connection to the MongoDB database using the appropriate database driver or ODM library. The connection is established using the connection string and configuration parameters such as the host, port, and authentication credentials.

3.3.4 CRUD Operations:

The backend performs CRUD operations on the database to store, retrieve, update, and delete application records. These operations include creating new application records when generated, retrieving specific applications based on filters, updating application information, and deleting records when necessary.

3.4 Testing and Quality Assurance

To ensure the reliability and quality of the Auto Applica application, testing and quality assurance practices are implemented. This includes the following:

3.4.1 Unit Testing:

Unit tests are written to verify the functionality of individual components, functions, and API endpoints. Test frameworks like Jest or Mocha can be used to write and execute the tests. Unit tests validate the correctness of application generation logic, data validation, and database interactions.

3.4.2 Integration Testing:

Integration tests are conducted to validate the interactions between different components and modules of the application. These tests ensure that the frontend and backend components work together seamlessly, API endpoints function correctly, and data flows as expected between the frontend and the database.

3.4.3 User Acceptance Testing (UAT):

User acceptance testing involves involving end-users to test the application and provide feedback. It helps identify any usability issues, user interface inconsistencies, or functional gaps that may have been missed during development. UAT ensures that the application meets the requirements and expectations of the target users.

3.4.4 Bug Fixing and Quality Assurance:

Throughout the implementation process, any identified bugs or issues are fixed promptly. Continuous quality assurance is performed to ensure the application functions as intended, meets the defined requirements, and delivers a satisfactory user experience. Code reviews, code quality checks, and performance optimizations may also be conducted to enhance the overall quality of the application.

CHAPTER 4

FEATURES OF APPLICATION AUTOMATOR

Figure 4.1 Features of a Web Application

The following are the features of the project discussed in brief:

1. Reduce complexity:

Many students face problems writing Applications and formal letters to the higher authority many of the time their application gets rejected because of the wrong indentation so with the help of our model they really don't have to write a complex application so the chances of their application getting rejected reduce effectively.

2. Easy to use:

The model is specially designed for an easy interface provided to all sorts of the user which may be educated or may be uneducated.

3. Time efficient:

In most cases, the process in which the application reaches the higher authority and after the results receiving the information back to the applicant takes a large amount of time but with the help of our system time duration will also decrease effectively.

4. Paperwork:

This model will also reduce the paperwork and maintain the history of all the applications and letters safe in the database.

5. Language:

This system will be available in two languages for now first is English and the second will be Hindi.

CHAPTER 5

INTERFACE OF APPLICATION AUTOMATOR

5.1. Login page

This login page includes username and password which will authenticate the user and take them to the profile of that user otherwise error occurs. This classify models for students, teacher, accountant and principal and provide the facility for login distinctly.

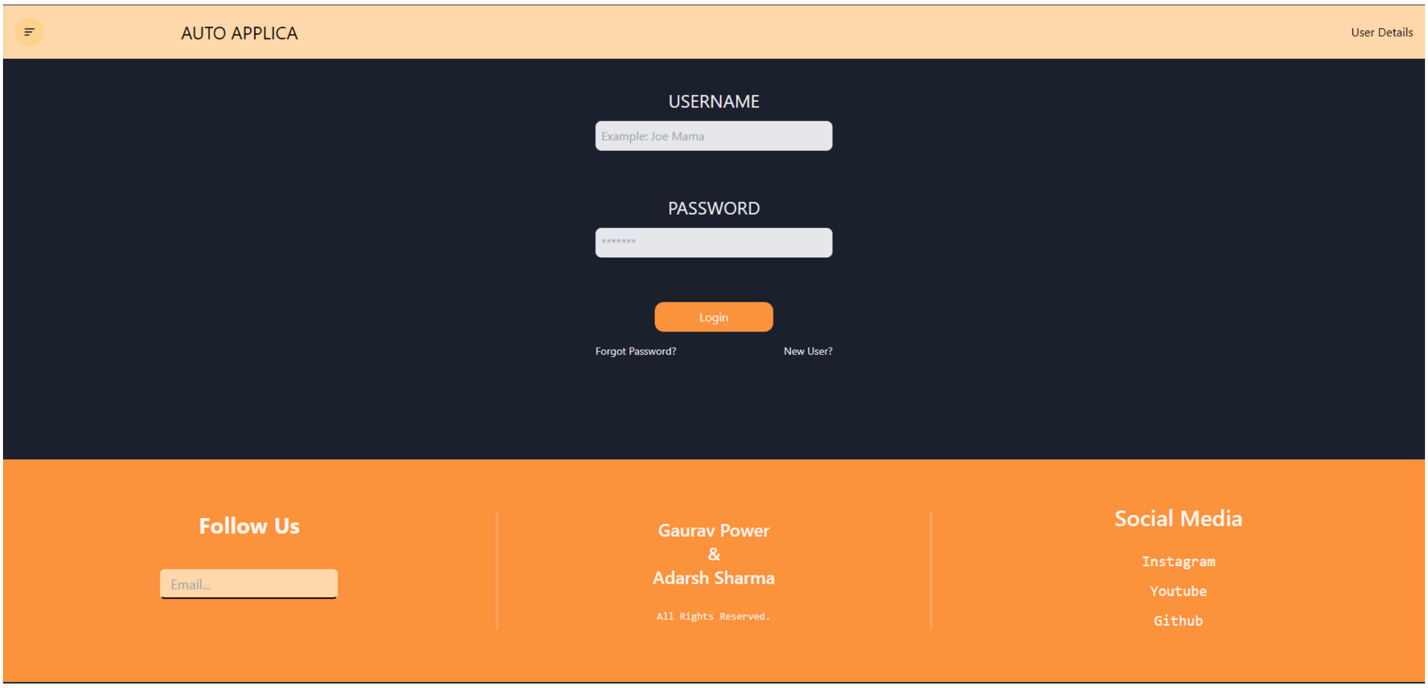
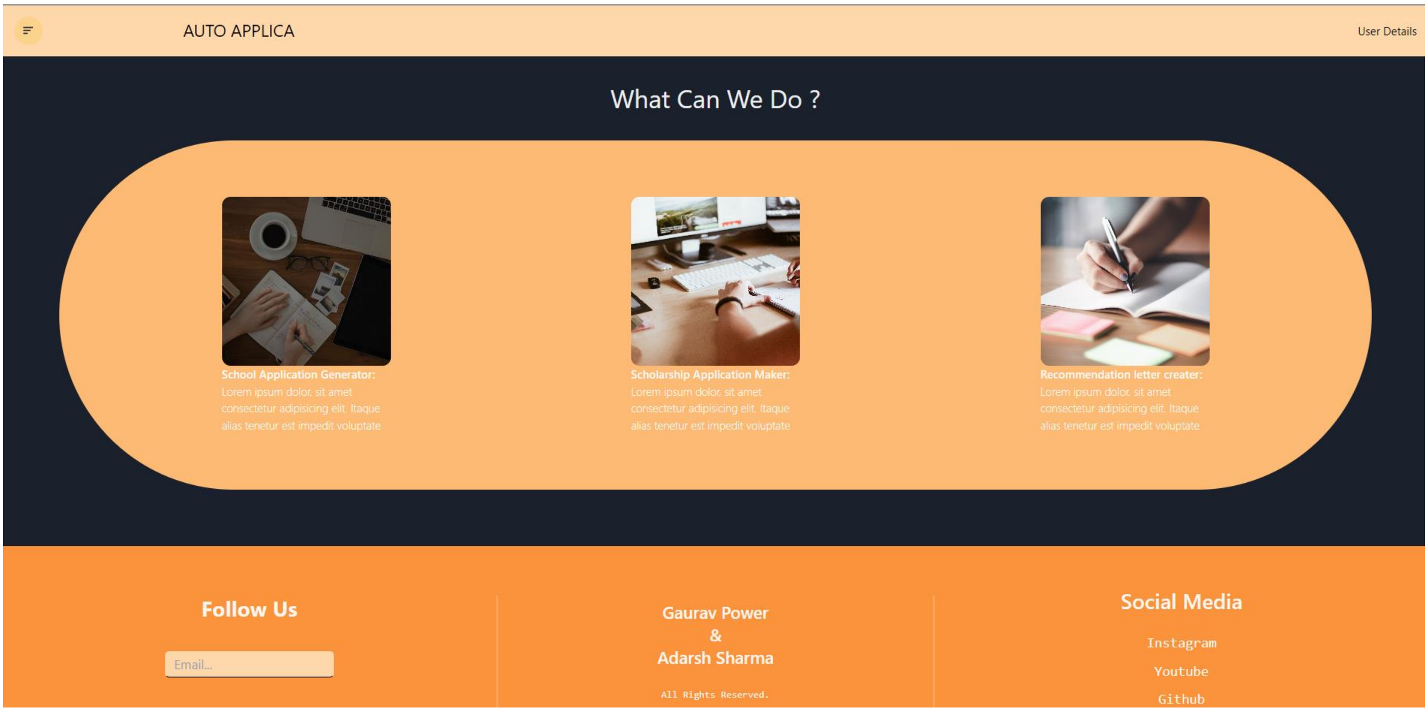


Figure 5.1 Login Page

5.2. Home Page

This interface represents the main profile page of the web application that provides user the detailed information about him where user can get options to find the status of application, history of applications and workspace to create application. This also provide the logout option with the support option which gives the user manual for using the web application.



What Can We Do ?

School Application Generator: Lorem ipsum dolor, sit amet consectetur adipisicing elit. Itaque alias tenetur est impedit voluptate

Scholarship Application Maker: Lorem ipsum dolor, sit amet consectetur adipisicing elit. Itaque alias tenetur est impedit voluptate

Recommendation letter creater: Lorem ipsum dolor, sit amet consectetur adipisicing elit. Itaque alias tenetur est impedit voluptate

Follow Us

Email...

Gaurav Power
&
Adarsh Sharma

All Rights Reserved.

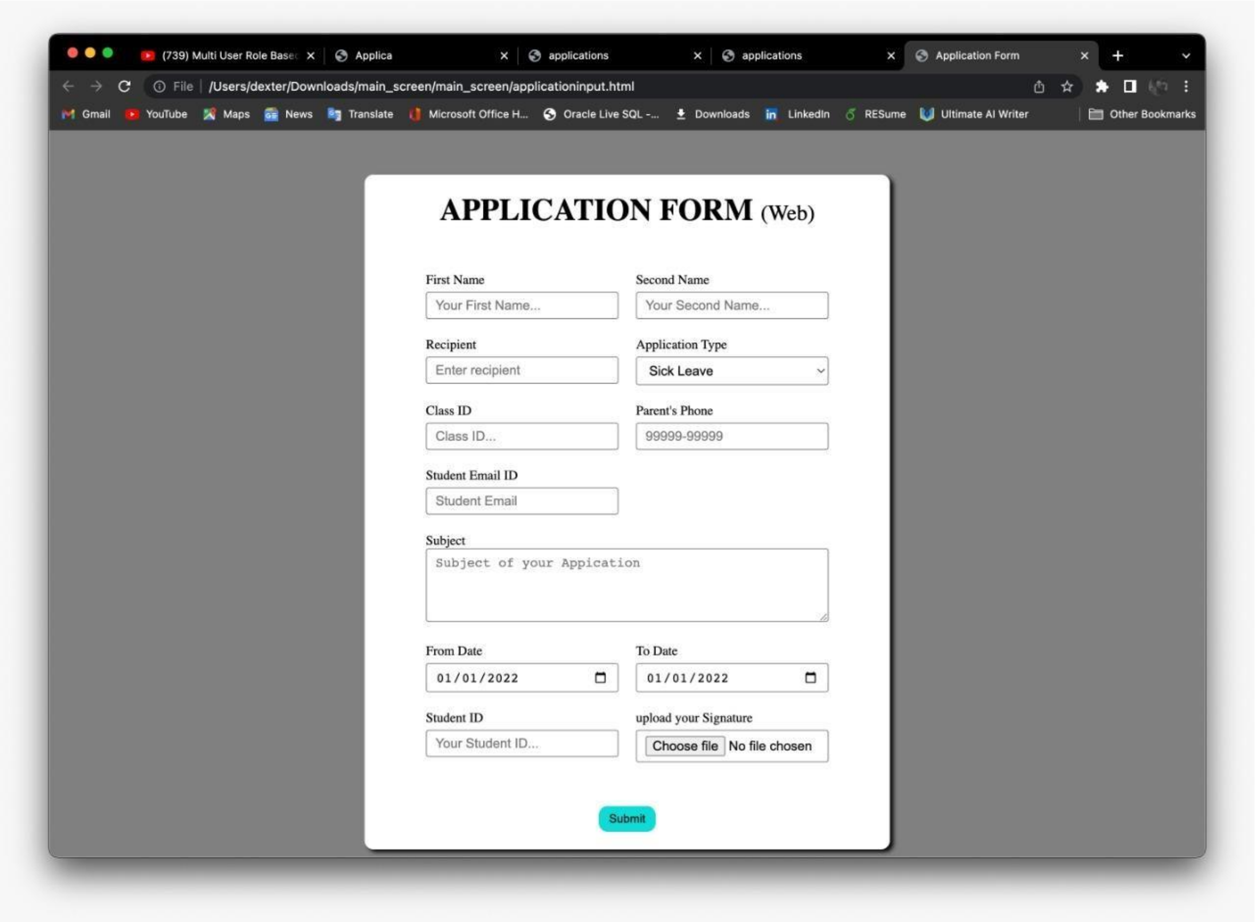
Social Media

Instagram
Youtube
Github

Figure 5.2 Home Page

5.3. Application form

This form is used to create an application according to the requirement of the user. This facilitates the user with a workspace where the user needs to fill some of the required details according to the application with appropriate reason and area for uploading signature is also provided. After the submission of this form, user will get preview of application which can be submitted and printed.



APPLICATION FORM (Web)

First Name Your First Name...	Second Name Your Second Name...
Recipient Enter recipient	Application Type Sick Leave
Class ID Class ID...	Parent's Phone 99999-99999
Student Email ID Student Email	
Subject Subject of your Appication	
From Date 01/01/2022	To Date 01/01/2022
Student ID Your Student ID...	upload your Signature Choose file No file chosen

Figure 5.3

CHAPTER 6

RESULT AND DISCUSSION

Chapter 6 focuses on the results and evaluation of the Auto Applica project. This chapter provides an overview of the outcomes achieved during the development process and evaluates the effectiveness and efficiency of the application in meeting its objectives. It includes the following key aspects:

Figure 6.1 Future of Online Documents

6.1 Implementation Results

In this section, the key results of the implementation phase are discussed. It highlights the successful completion of the frontend and backend development, integration of the required technologies (React.js, Node.js, Express.js, and MongoDB), and the utilization of CSS Tailwind for styling the user interface. The section also addresses any challenges or limitations encountered during the implementation process and the strategies employed to overcome them.

6.2 Functionality Evaluation

The functionality of Auto Applica is evaluated in this section. It assesses how well the application performs its intended tasks and meets the requirements of the target users, which are college and school students. The evaluation involves testing the various features and functionalities of the application, such as the ability to generate different types of applications (e.g., sick leave application), form validation, and the accuracy of the generated applications. User feedback and observations from the testing phase can be used to identify areas of improvement and refine the application's functionality.

6.3 Performance Evaluation

This section focuses on evaluating the performance of the Auto Applica application. Performance metrics, such as response time, scalability, and resource utilization, can be measured and analyzed to determine how well the application handles user requests and scales under different loads. Tools and techniques like load testing and performance profiling can be employed to gather data and assess the performance characteristics of the application. The evaluation results help identify any performance bottlenecks and guide optimization efforts, if required.

6.4 User Experience Evaluation

User experience (UX) evaluation is crucial to determine how well the application meets the usability and user satisfaction requirements. This evaluation can be conducted through various methods, including user surveys, interviews, and usability testing. Feedback from users regarding the user interface, ease of use, navigation, and overall experience can provide valuable insights for improving the application's UX. The evaluation results can inform design enhancements and feature adjustments to enhance the user experience.

6.5 Security Evaluation

Security is a critical aspect of any web-based application. In this section, the security measures implemented in Auto Applica are evaluated. This includes assessing the effectiveness of authentication and authorization mechanisms, data protection during transmission and storage, and protection against common security vulnerabilities such as cross-site scripting (XSS) and SQL injection. Vulnerability scanning tools and manual security assessments can be employed to identify any security weaknesses and ensure that appropriate measures are in place to safeguard user data and application integrity.

6.6 Limitations and Future Enhancements

This section discusses the limitations and potential areas for improvement in Auto Applica. It reflects on any constraints or challenges faced during the project implementation and highlights the scope for future enhancements. The limitations may include areas such as performance optimization, additional application types or templates, support for multiple languages, or integration with other platforms or systems. Suggestions and recommendations for addressing these limitations and expanding the application's functionality can be provided.

CHAPTER 7

CONCLUSION AND FUTURE ASPECTS

Chapter 7 concludes the report on the Auto Applica project and provides an overview of the project's achievements, contributions, and future prospects. It includes the following key aspects:

7.1 Summary of Project

This section provides a concise summary of the Auto Applica project, highlighting its objectives, scope, and key features. It recaps the development process, including the technologies used, the implementation phases, and the challenges faced during the project.

7.2 Impact and Benefits

This section explores the impact and benefits of the Auto Applica application. It discusses how the application addresses the needs of college and school students by streamlining the process of generating applications. The benefits can include time savings, increased accuracy in application generation, and improved convenience for both students and educational institutions. The section may also touch upon the potential positive outcomes, such as reducing paperwork, enhancing administrative efficiency, and improving communication between students and institutions.

7.3 Lessons Learned

The lessons learned section reflects on the insights and experiences gained throughout the project. It discusses the challenges encountered during the development process and the strategies employed to overcome them. Additionally, it highlights the valuable lessons learned in terms of project management, technology implementation, user feedback, and collaboration. This section provides valuable insights for future projects and serves as a reference for continuous improvement.

7.4 Future Work

In addition to the previously mentioned future work, an important expansion plan for the Auto Applica application is to make it available for educational institutions. This involves customizing the application to cater to the specific needs and requirements of institutions, enabling them to streamline their internal processes and enhance administrative efficiency. The following aspects can be considered as part of this future work:

7.4.1 Institution-Specific Customization:

To make Auto Applica suitable for institutions, customization options can be introduced. This includes providing configuration settings or an administration panel that allows institutions to define their own application types, templates, and approval workflows. Institutions can tailor the application to align with their internal policies and requirements, ensuring a seamless fit into their existing systems and processes.

7.4.2 User Management and Access Control:

Institution-specific customization requires implementing robust user management and access control mechanisms. This involves developing features such as user roles, permissions, and authentication methods tailored to the institution's organizational structure. Administrators can manage user accounts, assign appropriate access levels, and monitor application generation activities within the institution.

7.4.3 Integration with Institutional Systems:

To further enhance the usefulness of Auto Applica for institutions, integration with existing institutional systems can be explored. This may include integration with student information systems (SIS), human resource management systems (HRMS), or notification systems. Integrations can facilitate seamless data exchange, automate data synchronization, and streamline information flow between Auto Applica and other institutional systems.

7.4.4 Analytics and Reporting:

Institutions often require analytical insights and reporting capabilities to monitor application trends, track usage statistics, and generate reports for administrative purposes. Future work can focus on incorporating analytics and reporting features into Auto Applica, allowing institutions to gather meaningful data and derive valuable insights for decision-making and process optimization.

7.4.5 Scalability and Performance Optimization:

As the application expands to serve multiple institutions, scalability and performance optimization become crucial. Future work should consider implementing strategies to ensure that Auto Applica can handle increasing user loads and data volumes without compromising performance. This may involve optimizing database queries, introducing caching mechanisms, and leveraging cloud-based infrastructure to support scalability requirements.

7.4.6 User Training and Support:

To ensure successful adoption of Auto Applica by institutions, user training and support resources should be developed. This includes creating documentation, tutorials, and training materials to guide administrators and end-users on how to effectively utilize the application. Additionally, a support system, such as a help desk or online forum, can be established to address user queries, provide assistance, and gather feedback for further improvements.

By focusing on these future work areas, the Auto Applica application can evolve into a comprehensive solution for educational institutions, providing them with a powerful tool to streamline their application generation processes and improve administrative efficiency. This expansion will contribute to the wider adoption of Auto Applica and its positive impact on educational institutions and their stakeholders.

7.5 Conclusion

The conclusion of Chapter 5 summarizes the Auto Applica project's outcomes and contributions. It highlights the successful completion of the project, the achieved objectives, and the positive impact on college and school students. The conclusion emphasizes the value and significance of the application in simplifying the process of generating different types of applications. It also expresses gratitude towards the project team, stakeholders, and users who contributed to the project's success.

REFERENCES/BIBLIOGRAPHY

REFERENCES:

[1] https://blog.airslate.com/automation-education-streamline-educational-processes/

[2] Alcatel-Lucent, (2019). Why Digital Transformation for Education? https://www.al-enterprise.com/

[3] Amit Dua, (2018). Smart education is more than just advanced learning methods.https://yourstory.com/2018/05/smart-educationadvanced-learning.

[4] https://elearningindustry.com/list-of- tools-you-need-automation-in-schools-and-universities

[5] Jon Duckett. (2012). HTML and CSS: Design and build websites

[6] Marijn Haverbeke. *(2011). Eloquent JavaScript.*

[7] Lynn Beighley. (2002).Headfirst PHP & MySQL.

[8] Abraham Silberschatz & S Sudarshan. (2009). Database System Concepts.

[9] Jon Duckett. (2020).Web Design With HTML, CSS, JavaScript and jQuery Set.

PAPER NAME	AUTHOR
gaurav panwar minor report.docx	Gaurav Panwar

WORD COUNT	CHARACTER COUNT
3878 Words	23284 Characters
PAGE COUNT	FILE SIZE
22 Pages	1.5MB
SUBMISSION DATE	REPORT DATE
May 29, 2023 4:26 PM GMT+5:30	May 29, 2023 4:26 PM GMT+5:30

● 1% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 1% Internet database
- Crossref database
- 0% Submitted Works database
- 0% Publications database
- Crossref Posted Content database

● Excluded from Similarity Report

- Bibliographic material
- Cited material
- Quoted material
- Small Matches (Less then 16 words)