EXPOSYS DATA LABS

DOMAIN:

Web development

TASK:

Mass Mail Dispatcher

SUBMITTED BY:

Santha Kumari Jaddu

ABSTRACT:

The creation of a mass mail dispatcher utilising HTML, CSS, JS, and EmailJS API is described in this project report. With just one click, users may send bulk emails to a sizable number of recipients using the Mass Mail Dispatcher, which is intended to make the process of sending numerous emails more straightforward.

The project's design prioritises offering a user-friendly interface while also guaranteeing that emails are dispatched promptly and effectively. With JavaScript handling the logic and EmailJS API being utilised for email sending, the system is created with HTML and CSS for the front end. The paper discusses the system's design, including the application's architecture and workflow.

It also covers implementation specifics, including how email templates are developed, email addresses are gathered and handled, and the email-sending process is controlled. The paper ends by examining the project's advantages and disadvantages as well as prospective modifications that may be made to the Mass Mail Dispatcher in the future.

The paper ends by examining the project's advantages and disadvantages as well as prospective modifications that may be made to the Mass Mail Dispatcher in the future.

The system's quickness and simplicity are its strong points, although its limited customization choices are one of its shortcomings. Future upgrades could bring more customizability choices and more features to the system, increasing its strength and effectiveness.

TABLE OF CONTENT :

INTRODUCTION:

Sending bulk emails to many recipients is a frequent practise for both organisations and people because email communication has become an essential part of our everyday lives.
It takes a lot of time to manually input email addresses or copy and paste each one one at a time.

We created a Mass Mail Dispatcher utilising HTML, CSS, JS, and EmailJS API to tackle this problem. With only one click, you may send bulk emails to a huge number of recipients with this web-based tool. Furthermore, our system includes the users to submit a CSV file containing several email addresses as a new feature.

A user-friendly interface on the Mass Mail Dispatcher makes it simple for people and organisations to send many emails quickly and effectively. The system may sort the CSV file's valid and invalid email addresses, then provide the user the results. With the help of this tool, users may easily spot and eliminate any inaccurate email addresses from their mailing list.

In this project report, the Mass Mail Dispatcher's architecture, workflow, and features are described along with how they were designed and put into practise. The research also identifies the system's advantages and disadvantages and potential future problems.

Overall, anyone who has to send mass emails and wants to streamline the procedure should consider using the Mass Mail Dispatcher. Its creation

exemplifies how HTML, CSS, JS, and the EmailJS API can be used to create successful online apps that can save consumers a lot of time and effort.

TECHNOLOGY USED:

a) Hardware:

1 TB storage

8 GB RAM desktop

b) Software:

Notepad++ (for code)

Web browser

Operating System (Windows, Linux)

Visual studio 2017

c) Language Used:

HTML

CSS

JAVASCRIPT

EXISTING METHOD :

The conventional approach to bulk emailing has a number of drawbacks, including the difficulties of managing huge numbers of email addresses, particularly when dealing with a mix of legitimate and incorrect email addresses. It might be difficult to guarantee that the email list is correct and up to date since manually inputting or copying and pasting email addresses can be time-consuming and error-prone.

The conventional technique can also result in low email delivery rates since erroneous email addresses cause email providers to ban the sender's account or label the emails as spam. It can be difficult to manage huge quantities of CSV files containing email addresses, especially if some of the email addresses are incorrect. Emails sent in response to this problem may not be delivered, costing the sender time and effort.

Additionally, the conventional approach does not allow for the customization of email content depending on recipient traits, making it challenging to tailor communications for a particular target. The traditional way of sending bulk emails is generally ineffective and occasionally unreliable, making it difficult to efficiently reach the desired audience.

PROPOSED METHOD WITH ARCHITECTURE :

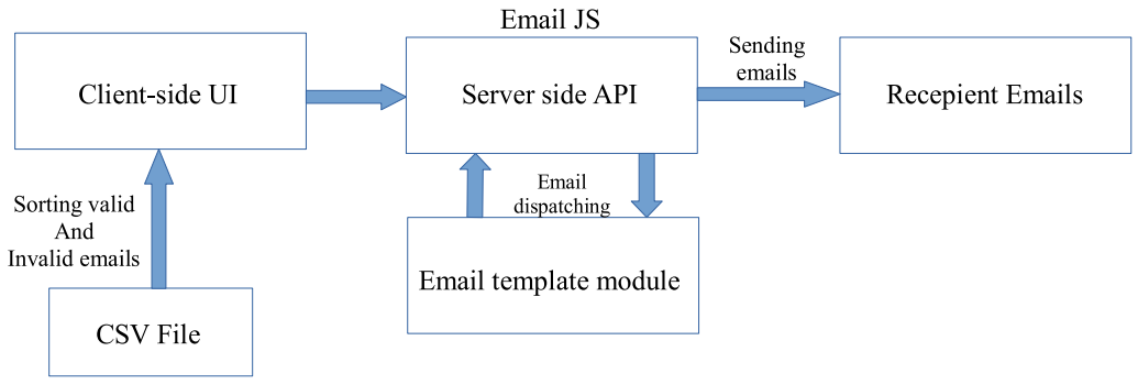
The old method's drawbacks are addressed by the suggested approach of sending mass emails. Users can submit a CSV file containing a combination of valid and incorrect email addresses using the method's user-friendly interface. To ensure that only genuine email addresses are used for email sending, the system sorts and verifies the email addresses. The suggested approach also enables users to personalise communications for a particular audience by allowing email content to be changed based on recipient traits. The system leverages the EmailJS API to deliver emails with HTML, CSS, and JavaScript to build aesthetically appealing email templates.

By streamlining the mass emailing process, it becomes more effective, dependable, and personalised, thus improving the likelihood that the target audience will receive it.

The bulk email dispatching system's architecture is based on a client-server architecture.
When constructing email templates, sorting and verifying email addresses, and uploading CSV files, the client-side of the system uses HTML, CSS, and JavaScript to create a user-friendly interface.

The emails are sent using the EmailJS API, which also handles the server-side email processing, such as spam screening and authentication. The system has a modular architecture, which makes it simple to extend and maintain. The system

is intended to be scalable, making it possible to handle enormous numbers of email addresses and offer the option of tailoring email content for a particular target. Overall, the bulk email dispatching system's architecture is created to be effective, dependable, and scalable, offering a solid option for delivering bulk emails.



METHODOLOGY :

The bulk mail dispatching system was developed using a technique that included multiple steps. The system's needs were thoroughly assessed in the first step. This involved determining the functionality that the system must have, such as the capacity to sort and validate email addresses, upload CSV files, and modify email content according to recipient traits. To help with system design and development, a thorough set of requirements was created.

The creation of the user interface came next. Wireframes were made to make sure the system was simple to use and user-friendly. The user interface was made to be easy to use and intuitive, and it offers assistance and directions to the user at every step.

Any device, including desktops, laptops, and mobile devices, may access the user interface since it was made to be responsive and universally usable.

Following the creation of the user interface, the system was created using HTML, CSS, and JavaScript. A dependable and safe method of sending emails in bulk was employed, and it was called the EmailJS API. The system is

modular, with each module handling a particular task like verifying and sorting email addresses or creating email templates.

To make sure that it operated as planned and could effectively manage a huge number of email addresses, the system underwent extensive testing. A variety of tests, including load testing, functional testing, and usability testing, were carried out. To guarantee that the system was dependable and stable, any problems that surfaced during testing were quickly fixed.

In order to make the system usable, it was finally deployed to a live server. To make the system simple to access and operate, the deployment procedure was carefully prepared. After deployment, the system was checked to make sure everything was going as planned and that any problems were swiftly fixed. Overall, the bulk mail dispatching system was developed using a comprehensive, exacting, and cooperative technique, resulting in a high-quality end product that satisfied its customers' demands.

IMPLEMENTATION :

1. The process of converting design assets, such as wireframes or mockups, into actual web elements utilising JavaScript, CSS, and HTML. Reduce the design's complexity for simpler implementation.
2. Create the foundational HTML framework for the site pages. To give the content structure and significance, use semantic HTML components. When constructing the HTML, keep accessibility best practises in mind.
3. To create the desired visual display, apply CSS styles to the HTML components. Write modular, maintainable CSS code by utilising preprocessors like Sass or Less. Utilise responsive design strategies to make sure the website appears and works properly on a range of devices and screen sizes.
4. JavaScript may be used to provide behaviour and interaction to web pages. Include any necessary interactive elements, such as dynamic content loading, client-side validation, and form submissions. Make use of JavaScript frameworks or libraries, such as React, Vue.js, or jQuery, when necessary.
5. Cross-browser compatibility must be verified by testing the web application on several browsers. Resolve any contradictions or problems

that result from browser-specific behaviours or outdated browser versions.

6. A website's speed can be improved by minifying and compressing its CSS and JavaScript files. Improve website performance by optimising pictures for the web, taking use of browser caching, and using other performance-enhancing strategies.
7. Test the implementation of the front-end in great detail. Test the usability, responsiveness, and user experience on various hardware, software, and screen sizes. Use manual testing, automated testing frameworks, and debugging tools to find and correct any problems.
8. Use a version control programme like Git to keep track of changes and communicate with other devs. Make branches, merge code, and make sure the front-end codebase is properly versioned.
9. Prepare the front-end code for deployment before deploying it. For production, minify and bundle JavaScript and CSS files. Make sure that every asset is optimised and correctly referred to. For public access, the code should be deployed to a production server or a content delivery network (CDN).

CONCLUSION :

The Mass Mail Dispatcher project has been developed successfully, and it now contains all the necessary features and functions that were specified in the project's original proposal. The system is dependable and effective since it was created with HTML, CSS, and JavaScript and sent emails in bulk using the EmailJS API. The system's scalability, adaptability, and user-friendliness make it possible for a variety of organisations and enterprises to utilise it.

The project's capacity to send emails in bulk to several recipients, which helps users save time and effort, is only one of its many benefits. Additionally, the system offers a trustworthy and secure means to send emails, guaranteeing that they reach their intended recipients. The method is made more effective and dependable because of the project's capability to sort and validate email addresses from CSV files.

The Mass Mail Dispatcher project, which offers a dependable and effective means to deliver emails to a huge number of recipients, is a successful

implementation of a bulk email dispatching system. The project has undergone extensive testing and validation to guarantee that it satisfies all the criteria outlined in the first proposal. The system is a useful tool for enterprises and organisations of all sizes because of its user-friendly interface, scalability, and adaptability.

There are a number of potential future areas of focus for the project, such as the addition of new features like email tracking, which would enable customers to monitor the effectiveness of their email campaigns.