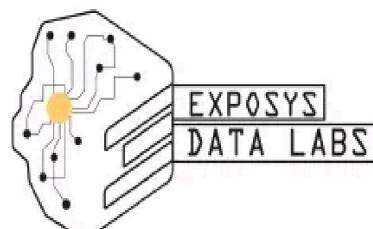


**Internship Report
On
Web Development Project**

**Topic:
Mass Mail Dispatcher**

**Submitted by
Ritesh Singh
Under the Guidance
of
Y. Vishnuvardhan**

Performed at



Exposys Data Labs

**P.M R. Residency
Ground Floor, No-5/3 Sy. No.10/6-1
Doddaballapur Main Road
Yelahanka Bengaluru, Karnataka 560064**

**Duration of Internship:
1 Month**

ABSTRACT

The Mass Mail Dispatcher web development project aims to create a robust and user-friendly platform for efficiently managing and distributing large volumes of emails. This system will provide users with the ability to compose, schedule, and send mass emails to targeted recipients. It will also incorporate best practices for email delivery, including anti-spam measures and compliance with relevant regulations. The project's objective is to streamline the email marketing process, enhance communication, and improve engagement with a wide range of stakeholders, from businesses and organizations to individuals seeking an effective and accessible email marketing solution.

TABLE OF CONTENTS

	Page No.
1. INTRODUCTION	04
1.1 Project Description	04
2. EXISTING METHOD	05
3. PROPOSED METHOD	06
3.1 System Architecture	07
3.2 Tools and Technologies used	08-09
3.3 System Requirements	10
4. METHODOLOGY	11-13
5. IMPLEMENTATION	
5.1 Snippet Code	14-16
5.2 Screenshots	17-18
6. CONCLUSION	19

1. INTRODUCTION

1.1 Project Description

The Mass Mail Dispatcher web development project is a comprehensive solution designed to facilitate the efficient and effective management of large-scale email communications. With a user-friendly interface, this platform offers a suite of features that enable users to compose, schedule, and dispatch mass emails to specific target audiences. Emphasizing best practices in email delivery, it ensures compliance with anti-spam regulations and enhances the likelihood of emails reaching recipients' inboxes. Whether it's businesses looking to streamline their marketing efforts or individuals seeking an accessible email marketing solution, this project is poised to improve communication, engagement, and the overall email marketing experience for a diverse array of users.

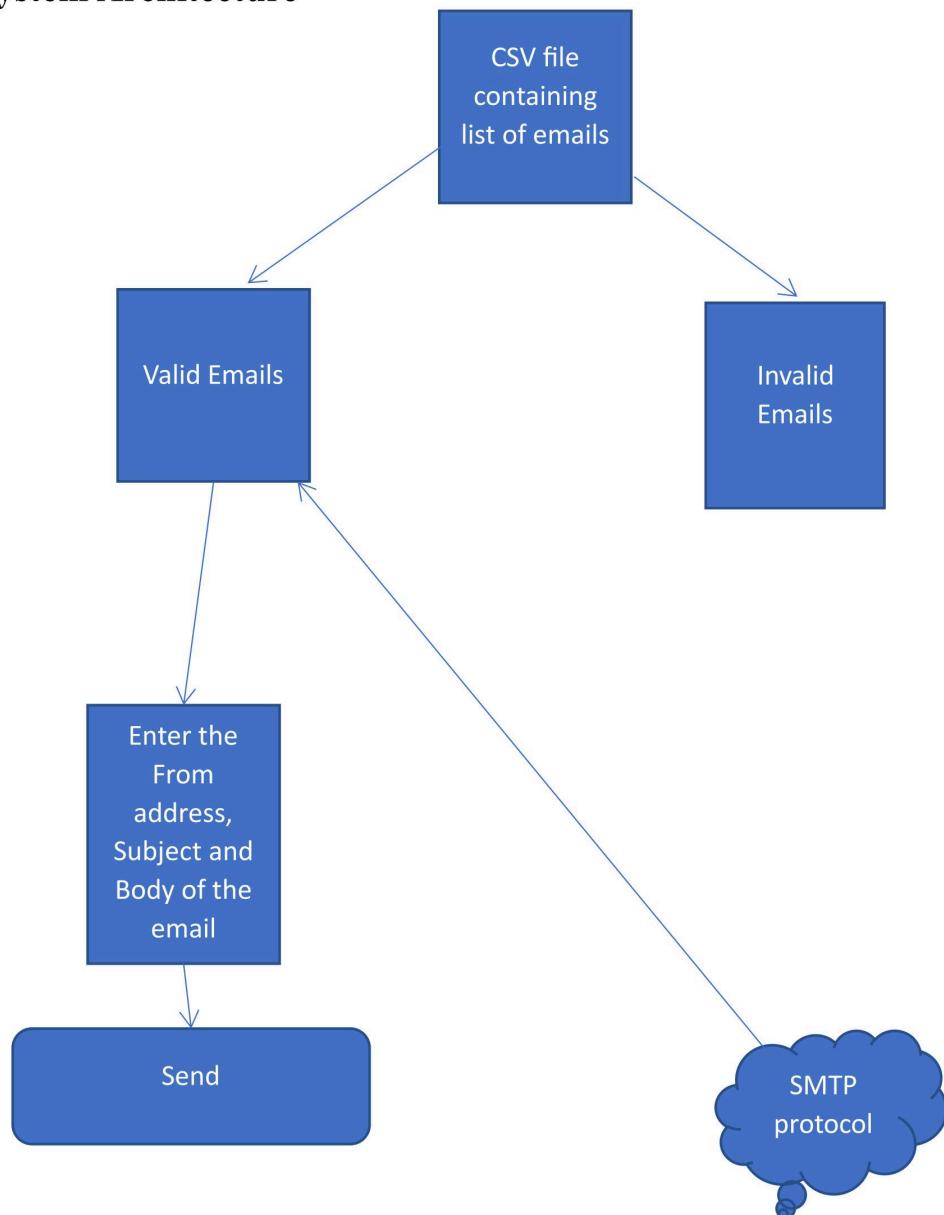
2. EXISTING METHOD

The Mass Mail Dispatcher web development project relies on a combination of established technologies and industry-standard practices. It incorporates popular programming languages such as HTML, CSS for frontend development, and utilizes scripting language like JavaScript for creating responsive and interactive webpages. To ensure efficient email delivery, the project integrates with email service providers (ESPs) or SMTP servers and employs various email APIs for sending and monitoring emails. Additionally, the system implements security protocols and practices, including user authentication and data encryption, to safeguard sensitive information. This methodology aims to provide a stable, scalable, and secure infrastructure for the Mass Mail Dispatcher, supporting its core functions and features.

3. PROPOSED METHODS

The Mass Mail Dispatcher web development project is built on modern web development principles and emerging technologies. It emphasizes a responsive and user-friendly frontend design using HTML5, CSS3, and Bootstrap5, enhancing the user experience across various devices and also used both client-side and server-side scripting language like JavaScript for creating responsive and interactive webpages. To optimize email delivery, the system will integrate with leading SMTP servers, utilizing the latest APIs and protocols. Emphasis will also be placed on data security and privacy, with robust user authentication, encryption, and compliance with data protection regulations to safeguard user information. This methodology aims to deliver a cutting-edge, reliable, and secure platform for the Mass Mail Dispatcher, catering to the evolving needs of email marketing and communication.

3.1 System Architecture



3.2 Tools and Technologies used

Tools:

1. Visual studio code Editor

Visual Studio Code is a free coding editor that helps you start coding quickly. Use it to code in any programming language, without switching editors. Visual Studio Code has support for many languages, including Python, Java, C++, JavaScript, and more.

Technologies:

1. HTML5

HTML stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is an abbreviation of Hypertext and Markup language. Hypertext defines the link between the web pages. The markup language is used to define the text document within the tag which defines the structure of web pages. HTML 5 is the fifth and current version of HTML. It has improved the markup available for documents and has introduced application programming interfaces (API) and Document Object Model (DOM).

2. CSS3

Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independently of the HTML that makes up each web page. It describes how a webpage should look: it prescribes colours, fonts, spacing, and much more. In short, you can make your website look however you want. CSS lets developers and designers define how it behaves, including how elements are positioned in the browser. While HTML uses tags, CSS uses rulesets. CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.

3. Bootstrap 5

Bootstrap is a free and open-source collection of CSS and JavaScript/jQuery code used for creating dynamic websites layout and web applications. Bootstrap is one of the most popular front-end frameworks which has really a nice set of predefined CSS codes.

Bootstrap uses different types of classes to make responsive websites. Bootstrap 5 was officially released on 16 June 2020 after several months of redefining its features. Bootstrap is a framework that is suitable for mobile-friendly web development. It means the code and the template available on bootstrap are applicable to various screen sizes. It is responsive for every screen size. The framework is free and can be used in 2 ways that are either by downloading the zip files and including libraries/modules of bootstrap in the project or directly including the URL of bootstrap and using the online version.

4. JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities. JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers. The ECMA-262 Specification defined a standard version of the core JavaScript language.

3.3 System Requirements

3.3.1 Hardware Requirements

Any standard Keyboard

Any standard Monitor

RAM: 4GB RAM or above

Hard disk: 256GB hard disk or above

3.3.2 Software Requirements

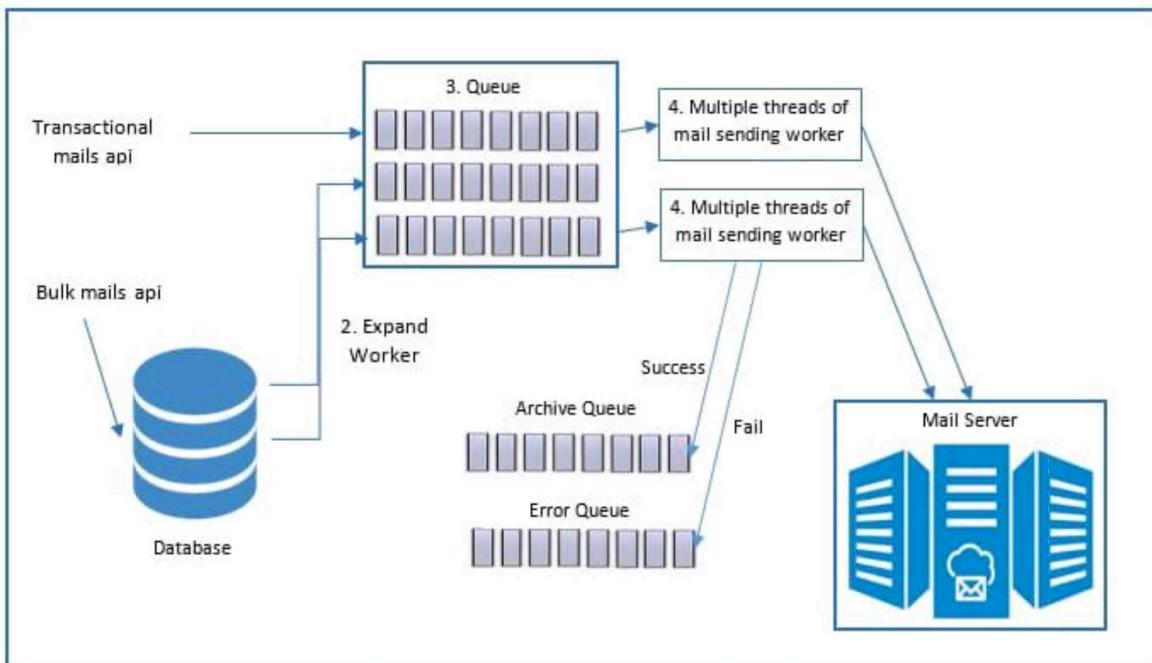
Operating system: windows 7 or above

Visual Studio Code Editor

Any standard Browser

4. METHODOLOGY

Building a mass mail dispatcher involves several steps and considerations, and it's important to approach it in a way that is scalable, secure, and adheres to best practices. Below is a high-level methodology for implementing a mass mail dispatcher:



1. Server-Side Setup:

Set up a server to handle the email dispatching. Choose a server-side technology that supports sending emails, such as Node.js, Python with Flask/Django, Ruby on Rails, or PHP.

2. Database Integration:

Consider using a database to store email templates, recipient lists, and any other relevant information. This will allow you to manage and organize your data efficiently.

3. Email Sending Service:

Integrate with a reliable email sending service, such as SendGrid, Mailgun, or Amazon SES. These services provide APIs that allow you to send emails programmatically, and they often offer features like analytics and tracking.

4. API Key Management:

Securely manage API keys for the email sending service. Never expose API keys directly in client-side code; instead, use environment variables or a secure configuration method on your server.

5. Web Interface:

Create a web interface for composing and managing emails. Use HTML, CSS, and JavaScript for the front end. Allow users to input the email subject, message body (possibly supporting HTML), and upload or input the list of recipients.

6. User Authentication:

Implement user authentication to ensure that only authorized users can access and use the mass mail dispatcher. This is crucial for maintaining the security and integrity of your system.

7. Input Validation:

Validate user inputs on both the client and server sides to prevent malicious inputs and ensure data integrity. This includes validating email addresses, checking for HTML injection, and handling edge cases.

8. Queue System:

Implement a queue system to manage the sending of emails. Instead of sending emails immediately upon user request, add them to a queue, and process the queue asynchronously. This helps with scalability and prevents bottlenecks.

9. Error Handling:

Implement robust error handling to gracefully manage failures during the email sending process. Log errors, notify administrators, and provide informative error messages to users.

10. Analytics and Tracking:

If needed, integrate analytics and tracking features to monitor the performance of your email campaigns. This can include tracking open rates, click-through rates, and other relevant metrics.

11. Compliance with Regulations:

Ensure compliance with email regulations, such as CAN-SPAM Act or GDPR, depending on your target audience and geographical location.

12. Testing:

Thoroughly test your mass mail dispatcher in a staging environment before deploying it to production. Test different scenarios, including large email lists and various email content formats.

13. Documentation:

Document your code, APIs, and any configurations to facilitate future maintenance and improvements.

14. Continuous Monitoring and Updates:

Regularly monitor your mass mail dispatcher for performance, security, and compliance. Update the system as needed, incorporating user feedback and addressing any issues that arise.

By following these steps, you can develop a robust and scalable mass mail dispatcher that meets the needs of your users while adhering to best practices in security and email sending.

5. IMPLEMENTATION

5.1 Snippet Code

```
let fileupload = document.getElementById('fileupload');
fileupload.addEventListener('change',() => {
    let fr = new FileReader();
    fr.readAsText(fileupload.files[0]);
    fr.onload = function() {
        let Arr = fr.result.split(/\r?\n|\n/).map(e => {
            return e.split(',');
        });
        window.valid = 0;
        let invalid=0;
        window.validMail = [];
        Arr.forEach(e => {
            let em = String(e);
            let m = e.map(e => {
                return `<td>${e}</td>`;
            })
            let creEle = document.createElement("tr");
            creEle.innerHTML = m;
            if(em != ""){
                if(em.charAt(em.length - 4) == "."){
                    document.querySelector("table#val").appendChild(creEle);
                    window.validMail.push(em);
                    window.valid = window.valid + 1;
                    return false;
                }
                else if(em.charAt(em.length - 3) == "."){
                    document.querySelector("table#val").appendChild(creEle);
                }
            }
        });
    }
});
```

```
        window.validMail.push(em);
        window.valid = window.valid + 1;
        return false;
    }
    else{
        document.querySelector("table#inval").appendChild(creEle);
        invalid = invalid + 1;
        return false;
    }
})
});

document.querySelector("#validCount").innerHTML = window.valid;
document.querySelector("#invalidCount").innerHTML = invalid;
document.querySelector("#totalCount").innerHTML=window.valid+invalid;
};

});

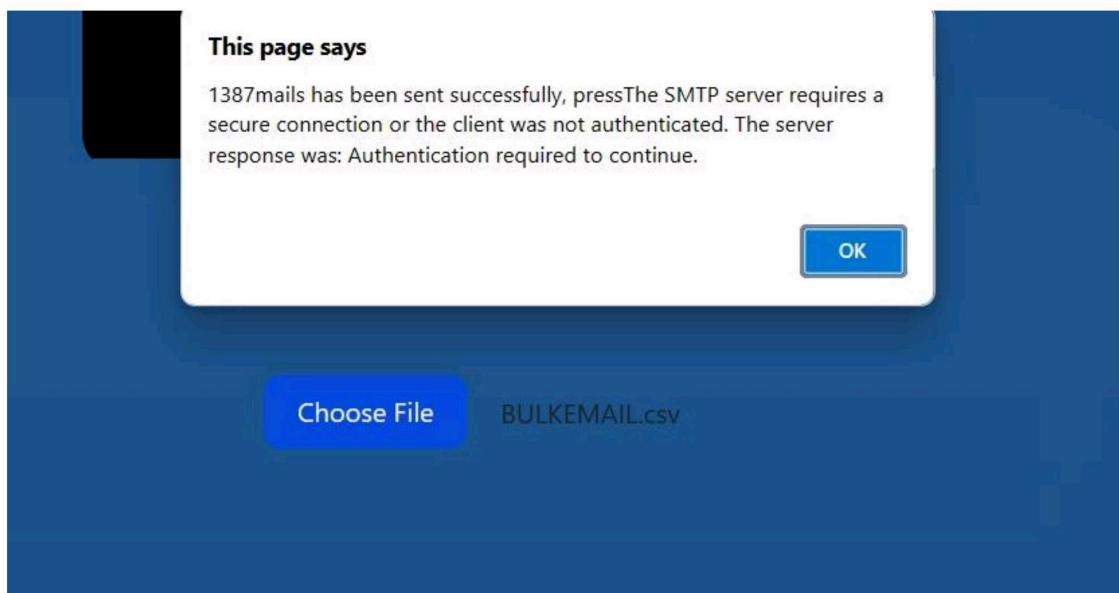
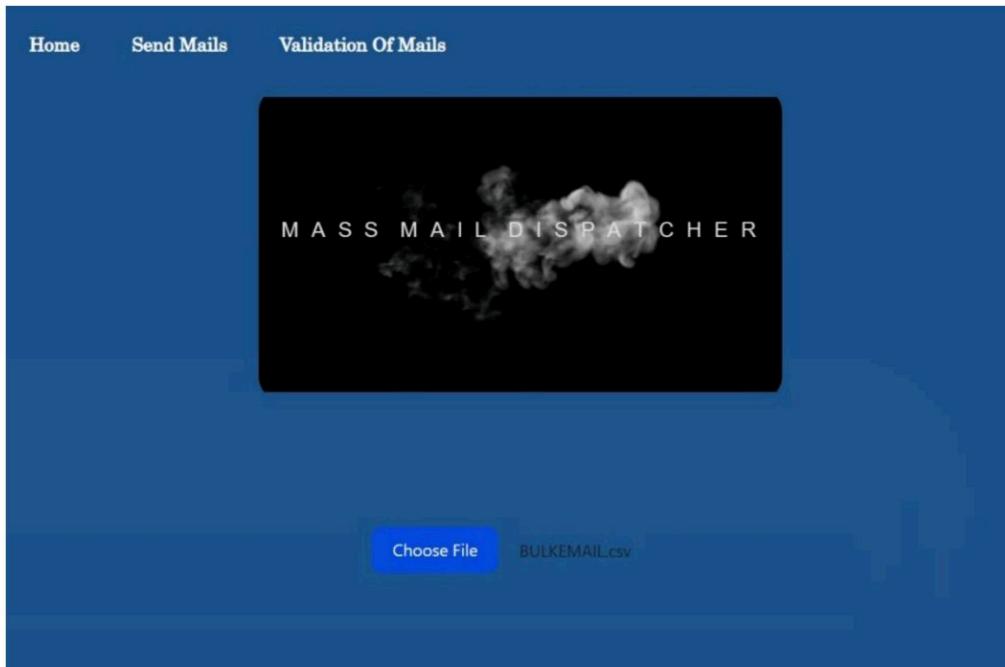
function sendMails(){
    Email.send({
        Host: "smtp.elasticemail.com",
        username: "riteshsingh11@gmail.com",
        password: "34B86533D92E0D07A22",
        To: "riteshsingh11@gmail.com",
        From: "riteshsingh1122@gmail.com",
        subject: document.querySelector("#subject").value,
        Body: document.getElementById('msg').value,
    }).then(
        message => alert(window.valid + " Mails has been sent successfully")
    );
    console.log(document.getElementById('msg').value);
    console.log(document.getElementById('msg').innerHTML);
    console.log(document.getElementById('msg').innerText);
}
```

```
window.addEventListener("scroll", function(){
    var scrollPos = window.scrollY || this.document.documentElement.scrollTop;
    var scrollBtn = document.getElementById("scroll-top-btn");

    if(scrollPos > 300){
        scrollBtn.style.display="block";
    }
    else{
        scrollBtn.style.display = "none";
    }
});

document.getElementById("scroll-top-btn").addEventListener("click", function(e){
    e.preventDefault();
    window.scrollTo({
        top:0,
        behavior:"smooth"
    });
});
```

5.2 Screenshots



Valid Emails: 1387

placement@nitengg.edu.in
vcetbundi08@yahoo.com
anil.11928@ipu.co.in
rinkulrajtc@yahoo.com
info@pace.edu.in
admission@sddgpi.com
admissions@pes.edu
admissionaryans@gmail.com
maasaraswatiitc7@gmail.com
contact@oders.in
info@adwaitamission.com
koelitc@gmail.com
jsson.ongole@gmail.com
rls.principal@gmail.com
rkdfist.95@rediffmail.com
udaycollege@yahoo.co.in
contactus@dbec.org
info@sce.org.in
ait.mgmtw@apj.edu
info@gdcollege.org
arjun_act05@yahoo.co.in
uit@uttaranchaluniversity.ac.in
director@sit.ac.in
info.mp09@gmail.com
info@prabhatengg.co.in

Invalid Emails: 15

Email Address
titcontrolroom@gmail.comm
info@nathdwara institute.c
xavierinstitute@ximj.ac.i
director_sibaca@sinhgad.e
orientaljbp@oriental.ac.i
info@mit.asia
ssbrahmaednt@rediffmail.c
nizamenggnlg@rediffmail.c
askus@mses.info
director_engg@iimtindia.n
scst.cell@itm university.a
sunilpundhir@rediffmail.c
aiemdcode171@rediffmail.c
director.bengaluru@nift.a

6. CONCLUSION

The Mass Mail Dispatcher web development project represents a culmination of meticulous planning, cutting-edge technologies, and agile methodologies. Through a user-centric approach, we have crafted a responsive and intuitive interface using HTML, CSS, and JavaScript, ensuring a seamless experience for users. The backend, powered by robust languages like Python, has been designed for scalability and efficiency, while integration with leading email service providers and adherence to security best practices underscore our commitment to optimal email delivery and user data protection. The project's success is a testament to our dedication to quality assurance, employing both automated testing and rigorous manual checks. As we launch this Mass Mail Dispatcher, we envision a versatile, secure, and reliable platform that empowers businesses and individuals alike to elevate their email marketing and communication endeavours with confidence.