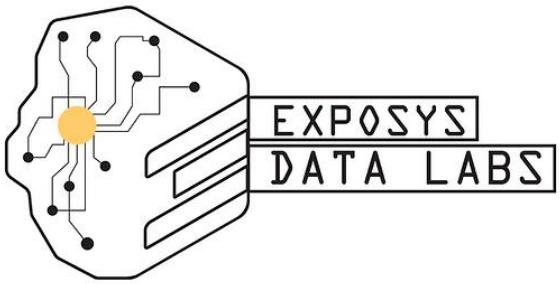


EXPOSYS DATA LABS

INTERNSHIP REPORT

On

"Mass – Mail Dispatcher"

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
In

INFORMATION SCIENCE & ENGINEERING

Submitted by
NANDINI KAPURCHAND RATHOD
Faculty Of Engineering And Technology
Jain University, Bengaluru – 562112

Under the guidance
of
Exposys Data Labs

Abstract

Mass-Mail Dispatcher makes the procedure of mass mailing quick and simple. Many people are first exposed to code through web development, yet little is known about the barriers beginners face in these formative experiences. In this article, we describe a study of undergraduate students enrolled in an introductory web development course taken by both computing majors and general education students. Using data collected during the initial weeks of the course, we investigate the nature of the syntax errors they make when learning HTML and CSS, and how they resolve them. This is accomplished through the deployment of open HTML, a lightweight web-based code editor that logs user activity.

Our analysis reveals that nearly all students made syntax errors that remained unresolved in their assessments, and that these errors continued weeks into the course. Approximately 20% of these errors related to the relatively complex system of rules that dictates when it is valid for HTML elements to be nested in one another. On the other hand, 35% of errors related to the relatively simple tag syntax determining how HTML elements are nested.

We also find that validation played a key role in resolving errors: While the majority of unresolved errors were A present in untested code, nearly all of the errors that were detected through validation were eventually corrected. We conclude with a discussion of our findings and their implications for computing education.

Table of Contents

1. Introduction

1. Introduction of domain

Web development is the work involved in developing a website for the Internet (World Wide Web) or an intranet (a private network). Web development can range from developing a simple single static page of plain text to complex web applications, electronic businesses, and social network services. A more comprehensive list of tasks to which Web development commonly refers, may include Web engineering, Web design, Web content envelopment, client liaison, client-side/server-side scripting, Web server and network security configuration, and e-commerce development. Among Web professionals, "Web development" usually refers to the main non design aspects of building Web sites: writing markup and coding.

Web development may use content management systems (CMS) to make content changes easier and available with basic technical skills. For larger organizations and businesses, Web development teams can consist of hundreds of people (Web developers) and follow standard methods like agile methodologies while developing Web sites. Smaller organizations may only require a single permanent or contracting developer, or secondary assignment to related job positions such as a graphic designer or information systems technician.

Web development may be a collaborative effort between departments rather than the domain of a designated department. There are three kinds of Web developer specialization: front-end developer, back-end developer, and full-stack developer. Front-end developers are responsible for behavior and visuals that run in the user browser, while back-end developers deal with the servers. Since the commercialization of the Web with Tim Berners-Lee developing the World Wide Web at CERN, the industry has boomed and has become one of the most used technologies ever.

2. Introduction of Project Internship

This is a Mass Mail Dispatcher where we send bulk mail to set of people at once. In dispatcher we can use CSV file formats which can contain bulk E-mail Address and checks if given any E-mail is valid or invalid.

2. Existing Method

Mass Mail Dispatcher is an existing method, but one without proper methodology. There is no adequate way to check whether the mail is valid and send the mass mail to those valid mail ids. We can't find a single solution to both of these problems. If we check whether the mail ID is valid or not, there is no way to send the mail to the valid ids in bulk. We put all these together and came up with the idea of checking the bulk mail IDs and sorting the CSV into valid and invalid mail IDs. After sorting the valid mail ids, they can send the mail to those valid mail ids.

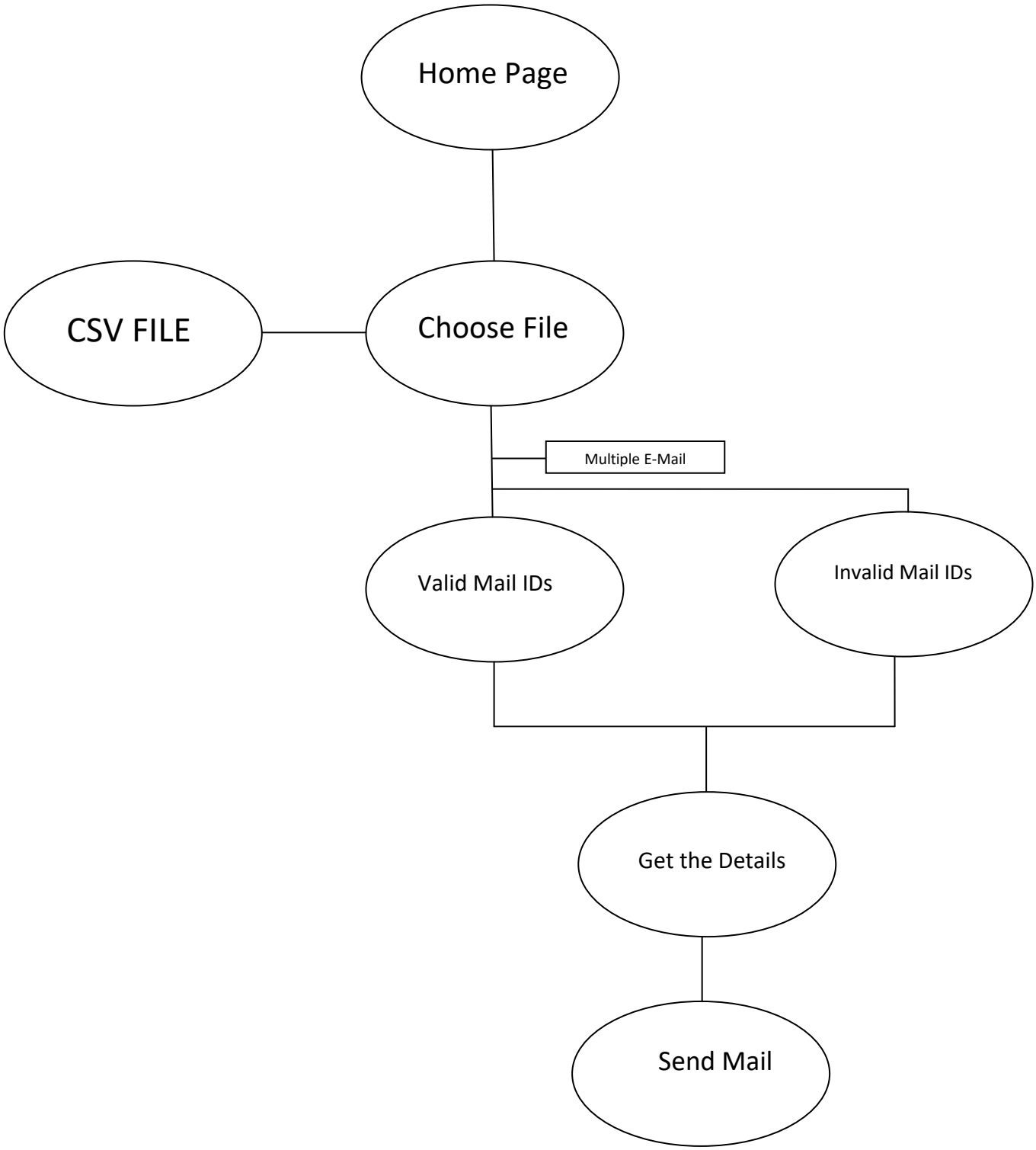
1. Single mail at a time sending.

2. Time taking as you need to send mails by composing mailing address every time.

3. There is no choice of add group, by add group one can sort out the address in a particular group order.

4. Sometimes lost of information while sending.

3. Proposed Method With Architecture



4. Methodology

We divided the work commonly and worked together. Frontend was done by two members and backend was done by a third member of our team.

4.1 Frontend:

In Frontend, we have three different pages
- Home Page
- Upload_CSV Page
- HTML

Home Page:
In Home Page, we have Title of the Project and two different sections namely Upload & verification and Send Mail. First click the Upload & Verification it will redirect into new page called upload_csv. After uploading and verifying the csv file. Go to next page called Send Mail.

Upload CSV Page:
In this page, there is two different buttons Choose file and Upload CSV. Click the choose file button and select the CSV file. After selecting the file click on the upload button. Once the button is clicking the CSV file is stored and verify the each and every mail if it's valid or Invalid. The Valid and Invalid Mail will display as two different tables in same page.

HTML:
HTML is a *markup language* that defines the structure of your content. HTML consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way.

4.2 Backend:

In Backend, we have two different pages
- CSS
- Emailjs

CSS

Used to add style to a web page by dictating how a site is displayed on a browser.

Emailjs

EmailJS helps to send emails using client-side technologies only. Supported email services, create an email template, and use one of our SDK libraries to trigger an email.

5. Implementation

Source Code

HTML Code:-

```
<!DOCTYPE html>
<html>
<head>
  <title>Send Mail</title>
  <script src="https://cdn.emailjs.com/dist/email.min.js"></script>
  <link rel="stylesheet" href="style.css">
  <link href="https://fonts.googleapis.com/css?family=Raleway:200,100,400" rel="stylesheet" type="text/css" />
</head>
<body>
  <h1>
    <span class="txt-rotate" data-period="2000"
      data-rotate='[ "   Hello !Myself", "Nandini Rathod ", "Welcome to", "  Mass-Mail", "DISPATCHER!!", " 
THANK-YOU !!" ]'></span>
  </h1>
  <p id="heading">EXPOSYS DATA LABS<br> MASS-MAIL DISPATCHER<br> NANDINI KAPURCHAND RATHOD</p>
  <form method="post">
    <label for="senderEmail">From :</label>
    <input type="email" id="senderEmail" name="senderEmail" required>
    <br><br><br>
    <label for="subject" style="margin-top: -2em;z-index: 22;position: absolute;">Subject:</label>
    <textarea id="subject" name="subject" required></textarea>

    <br><br><br><br>
    <label for="csvFile">CSV File:</label>
    <input type="file" id="csvFile" name="csvFile" accept=".csv" required>
    <br><br>
    <label for="message">Message:</label>
    <textarea id="message" name="message" required></textarea>
    <br><br>
    <input type="button" value="Send Emails" style="font-weight: bold;" onclick="sendEmails()">
    <br>
    <div style="display:flex">
      <div>
        <p style="color: #393E41;">Valid Emails: <span id="validEmailCount"></span></p>
        <div id="validEmails" style="float: left"></div>
      </div>
      <div style="margin-left: 15px">
        <p style="color: #393E41;">Invalid Emails: <span id="invalidEmailCount"></span></p>
        <div id="invalidEmails" style="float: left"></div>
      </div>
    </div>
  </form>
```

```
<span><img src="Images/boy.png" alt="" class="img1"></span>
<img src="Images/cloud.png" alt="" class="img2">
<div class="word"></div>

<script type="text/javascript">
  (function () {
    emailjs.init("5EpzE1X9S3UEAOSn1");
  })();

  function sendEmails() {
    var senderEmail = document.getElementById("senderEmail").value;
    var message = document.getElementById("message").value;
    var subject = document.getElementById("subject").value;
    var validEmails = [];
    var invalidEmails = [];
    var file = document.getElementById("csvFile").files[0];
    var reader = new FileReader();
    reader.readAsText(file);
    reader.onload = function (event) {
      var csv = event.target.result;
      var lines = csv.split('\n');
      for (var i = 0; i < lines.length; i++) {
        var email = lines[i].trim();
        var emailRegex = /^[\w-\.]+@([\w-]+\.)+[\w-]{2,3}$/
          ;
        if (emailRegex.test(email)) {
          validEmails.push(email);
        } else {
          invalidEmails.push(email);
        }
      }

      for (var j = 0; j < validEmails.length; j++) {
        var templateParams = {
          to_name: validEmails[j],
          from_name: senderEmail,
          message_html: message,
          subject_html: subject
        };

        emailjs.send('service_13s3n3a', 'template_huwrmzf', templateParams)
          .then(function (response) {
            console.log("SUCCESS", response);
          }, function (error) {
            console.log("FAILED", error);
          });
      }

      alert("Emails sent to valid email addresses.");
    };
  }
```

```
var TxtRotate = function (el, toRotate, period) {
  this.toRotate = toRotate;
  this.el = el;
  this.loopNum = 0;
  this.period = parseInt(period, 1) || 1000;
  this.txt = '';
  this.tick();
  this.isDeleting = false;
};

TxtRotate.prototype.tick = function () {
  var i = this.loopNum % this.toRotate.length;
  var fullTxt = this.toRotate[i];

  if (this.isDeleting) {
    this.txt = fullTxt.substring(0, this.txt.length - 1);
  } else {
    this.txt = fullTxt.substring(0, this.txt.length + 1);
  }

  this.el.innerHTML = '<span class="wrap">' + this.txt + '</span>';

  var that = this;
  var delta = 300 - Math.random() * 100;

  if (this.isDeleting) { delta /= 2; }

  if (!this.isDeleting && this.txt === fullTxt) {
    delta = this.period;
    this.isDeleting = true;
  } else if (this.isDeleting && this.txt === '') {
    this.isDeleting = false;
    this.loopNum++;
    delta = 500;
  }

  setTimeout(function () {
    that.tick();
  }, delta);
};

window.onload = function () {
  var elements = document.getElementsByClassName('txt-rotate');
  for (var i = 0; i < elements.length; i++) {
    var toRotate = elements[i].getAttribute('data-rotate');
    var period = elements[i].getAttribute('data-period');
    if (toRotate) {
      new TxtRotate(elements[i], JSON.parse(toRotate), period);
    }
  }
  var css = document.createElement("style");
  css.type = "text/css";
```

```
    css.innerHTML = ".txt-rotate > .wrap { border-right: 0.08em solid #666 }";
    document.body.appendChild(css);
  };
 </script>
 <script type="text/javascript">
  document.getElementById("csvFile").addEventListener("change", function () {
    var validEmails = [];
    var invalidEmails = [];

    var file = document.getElementById("csvFile").files[0];
    var reader = new FileReader();
    reader.readAsText(file);
    reader.onload = function (event) {
      var csv = event.target.result;
      var lines = csv.split('\n');
      for (var i = 0; i < lines.length; i++) {
        var email = lines[i].trim();
        var emailRegex = /^[\w-\.]+@([\w-]+\.)+[\w-]{2,3}$/
          ;
        if (emailRegex.test(email)) {
          validEmails.push(email);
        } else {
          invalidEmails.push(email);
        }
      }

      document.getElementById("validEmails").innerHTML = validEmails.join("<br><br>");
      document.getElementById("invalidEmails").innerHTML = invalidEmails.join("<br><br>");
      document.getElementById("validEmailCount").innerText = "(" + validEmails.length + ")";
      document.getElementById("invalidEmailCount").innerText = "(" + invalidEmails.length + ")";
    };
  });


 </script>

</body>

</html>
```

CSS Code:-

```
  font-family: Arial, sans-serif;
  background-color: #99c9ed;
  font-family: Verdana, Geneva, Tahoma, sans-serif;

  background-size: cover;
  position:relative;
  color: #333;
  overflow-y: hidden;
  z-index: 2;
}



/* Center the main heading and add margin at the top */
#heading {
color:
#0e0e0e;
margin-left: .5em;
font-size: 2em;
font-weight: bolder;
position: fixed;
margin-top: .2em;

}


/* Style the form container */
form {
  max-width: 600px;
  margin-left: 52em;
  margin-top: 1em;
  /* From https://css.glass */
  background: rgb(239, 235, 235);
  border-radius: 16px;
  box-shadow: 0 4px 30px rgba(0, 0, 0, 0.1);
  backdrop-filter: blur(2.7px);
  -webkit-backdrop-filter: blur(2.7px);
  border: 1px solid rgba(255, 255, 255, 0.03);
  padding: 20px;
  /* border-radius: 10px; */
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
  position: relative;
}

/* Style form labels */
label {
  display: block;
  margin-bottom: 10px;
  font-weight: bold;
  color: #575761;
}

/* Style text inputs and text areas */
input[type="email"],
textarea {
  width: 95%;
  padding: 10px;
  margin-bottom: 10px;
  border: 1px solid #767a7d;
  border-radius: 5px;
  font-size: 16px;
  color: #766f6f;
  background: rgba(187, 179, 179, 0.1);
}

  /* Style text inputs and text areas */
  #subject{
    width: 89%;
    padding: 10px;
    margin-bottom:-1em;
    border: 1px solid #767a7d;
    border-radius: 5px;
    font-size: 16px;
    color: #766f6f;
    z-index: 22;
    position: absolute;

    background: rgba(187, 179, 179, 0.1);
  }

/* Style file input */
input[type="file"] {
  margin-bottom: 12px;
  color: #393E41;

}

/* Style form submit button */
input[type="button"] {
  display: block;
  width: 100%;
  padding: 10px;
  background-color: #C5C3C6;

  border: none;
  border-radius: 5px;
```

```
  color: #393E41;
  font-size: 16px;
  cursor: pointer;
  transition: background-color 0.3s;
}

/* Style submit button on hover */
input[type="button"]:hover {
  color: #e6e6e6;
  background-color:#65696d;

}

/* Style container for valid and invalid emails */
#validEmails {
  margin-top: 1px;
}
#invalidEmails {
  margin-top: 1px;

}

/* Style paragraphs within container for valid and invalid emails */
#validEmails p,
#invalidEmails p {
  font-weight: bold;
  color: #393E41;
}


/* Style divs within container for valid and invalid emails */
#validEmails div {
color: #393E41;
}

#invalidEmails div {
color: #393E41;
}

#message{
  color:#393E41;
}
```

```
#senderEmail{
  color:#393E41;
}

/* Style text inputs and text areas */
input[type="email"]:focus,
textarea:focus {
  outline: none !important;
  border-color: #393d40;

}
#validEmails, #invalidEmails {
  height: 150px;
  overflow: auto;
}


.img1{
height:380px;
margin-top: -25em;
position: fixed;
}

.img2{
height:200px;
margin-top:-32em;
position: absolute;
margin-left: 11.3em;
}

h1,h2 {
font-weight: 200;
margin: 0.4em 0;
z-index: 12;
position: absolute;
margin-top: 15em;
margin-left: 12em ;
}
h1 { font-size: 18px; }
h2 {
color: #888;
font-size: 18px;
}
```

CSV Code:-

```
var file = document.getElementById("csvFile").files[0];
    var reader = new FileReader();
    reader.readAsText(file);
    reader.onload = function (event) {
      var csv = event.target.result;
      var lines = csv.split('\n');
      for (var i = 0; i < lines.length; i++) {
        var email = lines[i].trim();
        var emailRegex = /^[\w-\.]+@([\w-]+\.)+[\w-]{2,3}$/
          ;
        if (emailRegex.test(email)) {
          validEmails.push(email);
        } else {
          invalidEmails.push(email);
        }
      }
```

JS Code:-

```
let upload = document.getElementByTd('upload');
upload.addEventListener('change',() => {
   let fr = new FileReader();
   fr.readAsText(upload.files[0]);
   fr.onload = function () {

      let Arr = fr.result.split(/\r?\n|\n/).map(e => {
         return e.spilt(',');
      });
      window.valNo = 0;
      let invalNo = 0;
      Window.valMail = [];
      Arr.forEach(e => {
         let em = String(e);
         let m = e.map(e => {
            return <td>${e}</td> ;
         })
         let creEle = document.createElement("tr"); // table row
         creEle.innerHTML = m;
         if (em != "") {// so that blank row will not be printed as well as counted
            // if (em.indexOf('@') !=0) {
            //    document.querySelector("table#val").appendChild(creEle);
            //    return false;
            // }
            if (em.charAt(em.length - 4) == ',') {
               document.querySelector("table#val").appendChild(creEle);
               Window.valMail.push(em);
               Window.valNo = Window.valNo + 1;
               return false;
```

```
            }
            else if (em.charAt(em.length - 3) == ',') {
                document.querySelector("table#val").appendChild(creEle);
                Window.valMail.push(em);
                Window.valNo = Window.valNo + 1;
                return false;
            }
            else {
                document.querySelector("table#inval").appendChild(creEle);
                invalNo = invalNo + 1;
                // console.log(creEle);
                return false;
            }
        }
    });

    document.querySelector('#valCount').innerHTML = Window.valNo;
    document.querySelector('#invalCount').innerHTML = invalNo;
  }
});
//--------------sending emails-------------

function sendEmails() {
  sendEmails.send({
    Host: "smtp.elasticemail.com",
    Username: "",
    Password:
    To:
    Form:
    Subject:
    Body:
  }).then(
    message => alert(window.valNo + "mails has been sent successfully, press " + message + " to continue")
  );
  console.log(document.getElementById('msg').value);
  console.log(document.getElementById('msg').innerHTML);
  console.log(document.getElementById('msg').innerText);
}
```

Open this link to find other codes:

Github link:- https://github.com/NandiniRathod02/Intern_Project01/tree/main

6. Conclusion

Becoming a web developer is more than just working in the browser. Web developers write applications, and you are on that path. I encourage you to continue learning HTML, JavaScript, CSS, and Python. But when you're comfortable, branch out to other languages and platforms. The more exposure and experience you have with other languages and platforms, the better you'll become as a developer.
