

EMAIL,WHATSAPP,MORMAL MESSAGE ON ONE PLATFORM

Submitted in partial fulfillment of the requirements for the award of Bachelor of Degree in Computer Science

By

DURGA DHARSHINI.K(Reg.No.39290023)

MATHUMITHA.R(Reg.No.39290065)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119, TAMILNADU

APRIL 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of DURGA DHARSHINI.K (Reg. No. 39290023) and MATHUMITHA.R (Reg. No. 39290065) who carried out the project entitled "EMAIL WHATSAPP,NORMAL MESSAGE ON ONE PLATFORM" under my supervision from November 2020

to April 2021.

INTERNAL GUIDE

DR.SELVI

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.

DECLARATION

I DURGA DHARSHINI.K hereby declare that the Project Report entitled EMAIL, NORMAL ,WHATSAPP MESSAGE ON ONE PLATFORM done by me under the guidance of Dr./Prof./Mr./Ms. DR.M.SELVI (Internal) is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in COMPUTER SCIENCE.

DATE: 08-04-2022

K.Durga dharshini

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean, School of Computing Dr. L. Lakshmanan M.E., Ph.D. , and Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr./Mr./Ms for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

EMAIL, WHATSAPP NORMAL MESSAGE ON ONE PLATFORM

ABSTRACT

The intention of this project is to build an automation tool using java for and sending mails ,WhatsApp message , normal message on one application.

We will be creating a workflow in scheduler option which queries the details , according to a given search criteria, and sends their details as whatsapp message ,normal message, or mails.

Now a days we have used many programs or application to do our work .like this we have used many programs for many different woks . in this time we have lost some work instead of using another program sincerely. For this situation we use a single platform to do our work fluently.

SL.NO	TITLE DESCRIPTION	PAGE NO
1.	INTRODUCTION	
	1.1 OVER VIEW OF PROJECT	5
2.	LITERATURE SURVEY	6
3.	AIM AND SCOPE PRESENTATION	
	3.1 AIM OF THE PROJECT	7
	3.2 SCOPE AND OBJECTIVE	9
	3.3 SYSTEM REQUIREMENTS	13
	3.3.1 SYSTEM REQUIREMENTS	16
	3.4 SOFTWARE USED	18
4.	EXPERIMENTAL OR MATERIAL METHODS	
	4.1 DESIGN METHODOLOGY	20
	4.1.1 EXSISTING SYSTEM	22
	4.1.2 PROPOSED SYSTEM	24
	4.2 MODULE DESCRIPTION	26
	4.3 ARCHITECTURE DIAGRAM	28
	4.4 SAMPLE OUTPUT	29
5	FUTURE ENHANCEMENT	30
6	SOURCE CODE	31

INTRODUCTION

1.1 OVER VIEW OF PROJECT

* The intention of this project is to build an automation tool using java for and sending mails ,WhatsApp message , normal message on one application. We will be creating a workflow in scheduler option which queries the details , according to a given search criteria, and sends their details as whatsapp message, normal message , or mails.

*node mailer,mailgun is an efficient platform which provides us with the features needed to accomplish this. It's a message, email, call and notification automation tool/platform. We'll be utilizing some of its features. The purpose of this project is to design an automation tool where we can send our mails ,message in normal or whatsapp consisting of text.

* Sending mails,message in WhatsApp or normal message is much easier because we spend most of the time on switching to another app can save us time and it also facilitates in user flexibility.

LITERATURE SURVEY

3. AIM AND SCOPE OF PRESENT INVESTIGATION

3.1 AIM OF THE PROJECT

To work on several programs using a single platform instead of using more programs.

3.2 OBJECTIVE

You will be creating a workflow in an scheduler option which queries the details, according to a given search criteria, and sends their details as WhatsApp message, normal message, or mails.

3.4 SYSTEM REQUIREMENTS

RAM	:	2 GB DDR RAM
Hard Disk	:	500 GB
Key Board	:	Standard IOS Keys
Monitor	:	14' color monitor
Mouse	:	3 Button mouse

3.5 SOFTWARE REQUIREMENTS

Operating System	:	MS-Windows
XP/7. Front End	:	HTML
Back End	:	JAVASCRIPT

4. EXPERIMENTAL OR MATERIAL METHODS

4.1 DESIGN METHODOLOGY

4.1.1 EXSISTING SYSTEM

The first part of the contact form deals with email notification using Nodemailer to send emails.

The second part of the contact form deals with sms notification.

The third part of the contact form deals with whatsapp notification.

- In third part of whatsapp notification we have used node-cron to schedule the job.

- To understand the schedule part one might refer to ---><https://crontab.guru/>

- However we can apply this scheduler in other parts of notification like email, sms as well.

- The other way to schedule the job would be to Setup Scheduled Events aka CRON Jobs with AWS Cloudwatch Events

4.1.2 PROPOSED SYSTEM

- * We will be creating a workflow in which queries the requested email data, according to a given search criteria, and sends their details to WhatsApp.
 - * Node mailer,mailer gun is an efficient platform which provides us with the features needed to accomplish this. It's a message, email, sending system.
 - * We'll be utilizing some of its features through this project.
 - * The node mailer for normal message is a pre-configured environment available through the mailer gun Console in which you can prototype sending outbound messages, replying to incoming messages, and configuring things like message delivery callbacks.
 - * IMAP is an email retrieval protocol which does not download the emails. It just reads them and displays them. This is very useful in low bandwidth condition.
- .

4.2 MODULE DESCRIPTION

1. `const express = require("express");`

Creates an Express application. The `express ()` function is a top-level function exported by the `express` module. `const express = require ('express') const app = express ()`
Methods `express.json ([options])` This is a built-in middleware function in Express. It parses incoming requests with JSON payloads and is based on `body-parser`.

2. `const bodyParser = require("body-parser");`

· This how to use `body-parser` in `express`: `const express = require('express'), app = express(), bodyParser = require('body-parser'); // support parsing of application/json type post data app.use(bodyParser.json()); //support parsing of application/x-www-form-urlencoded post data app.use(bodyParser.urlencoded({ extended: true }));` Link.

3. `const webpush = require("web-push");`

`const webpush = require('web-push');` // **new** To subscribe push messages, we need to pass VAPID keys. We can generate VAPID keys as below. `const webpush = require('web-push'); console.log(webpush.generateVAPIDKeys());` // **new**.

4. `const expHbs = require("express-handlebars");`

const expressHbs = require ('express-handlebars'); app.engine ('handlebars', expressHbs.engine ({ layoutsDir: 'views/layouts/', defaultLayout: null, extname: 'handlebars' })); app.set ('view engine', 'handlebars'); app.set ('views', 'views'); // optional.

5. `const nodemailer = require("nodemailer");`

Include the nodemailer module in the code using `require ('nodemailer')`. Use `nodemailer.createTransport ()` function to create a transporter who will send mail. It contains the service name and authentication details (user and password).

6. `const schedule = require('node-schedule')`

Remember - in JavaScript - 0 - January, 11 - December. **const schedule = require ('node-schedule');** **const** date = new Date (2012, 11, 21, 5, 30, 0); **const** job = **schedule**.scheduleJob (date, function () { console.log ('The world is going to end today.')}); But this code runs the task every minute, I just want it to run the task only once at the ...

4.3 ARCHITECTURE DIAGRAM

Sender(sending mail)

||

Node mailer(which is used for sending mail to the mailer app)

||

Receiver(receiving mail send from the node mailer)

Sender(sending message

||

Twilio(which is used for sending message)

||

Receiver(receiving message)

Sender(sending whatsapp message)

||

Wbm(which is used for sending whatsapp message)

||

Receiver(receiving whatsapp message)

4.4 SAMPLE OUTPUT

Notification

📍 Meet me at...

☎ (0612) 2288332

✉ test@gmail.com

To Email Notification

Name

Company

Email Address

Phone Number

Message

SUBMIT FOR EMAIL NOTIFY

To Message Notification On Your Phone

Phone Number

Format:917003666622

Message

SUBMIT FOR SMS NOTIFY

To Whatsapp Notification On Your

Phone Number

5.FUTURE ENHANCEMENT

We have to use a twilio for whatsapp message method to send a message on one platform.in this method we have to use java script code to develop a platform.

6. SOURCE CODE

FOR SENDING MAIL

```
const express = require("express");
const bodyParser = require("body-parser");
const webpush = require("web-push");
const exphbs = require("express-handlebars");
const path = require("path");
const nodemailer = require("nodemailer");
const twilio = require("twilio");
const wbm = require("wbm");
const schedule = require('node-schedule')
const cron = require('node-cron');
```

```
require("dotenv").config();
```

```
const app = express();
```

```
console.log(process.env);
```

```
// View engine setup
```

```
app.engine("handlebars", exphbs());
```

```
app.set("view engine", "handlebars");
```

```
// Static folder
```

```
app.use("/public", express.static(path.join(__dirname,  
"public")));
```

```
//app.use(express.static(path.join(__dirname,"views")))
```

```
// Body Parser Middleware
```

```
app.use(bodyParser.urlencoded({ extended: false }));
```

```
app.use(bodyParser.json());
```

```
app.get("/", (req, res) => {
```

```
//
res.render('layouts/main.handlebars');//res.render('contact')
res.render("layouts/main.handlebars"); //
//res.render('contact')
});
```

```
app.post("/send", (req, res) => {
  const output = `
    <p>You have a new contact request</p>
    <h3>Contact Details</h3>
    <ul>
      <li>Name: ${req.body.name}</li>
      <li>Company: ${req.body.company}</li>
      <li>Email: ${req.body.email}</li>
      <li>Phone: ${req.body.phone}</li>
    </ul>
    <h3>Message</h3>
    <p>${req.body.message}</p>
  `;
  ;
```

```
// create reusable transporter object using the default SMTP
transport

let transporter = nodemailer.createTransport({
  // host: 'mail.YOURDOMAIN.com',
  // port: 587,
  // secure: false, // true for 465, false for other ports
  service: "gmail",
  auth: {
    user: "sinhaanwsha16@gmail.com", // generated ethereal
    user
    pass: process.env.API_KEY, // generated ethereal
    password...to allow this 1st allow less secure apps in ur email
    account via this url----->
    https://myaccount.google.com/lesssecureapps
  },
  // tls:{
  //   rejectUnauthorized:false
  // }
});

// setup email data with unicode symbols

let mailOptions = {
```

```
    from: "sinhaanwsha16@gmail.com", //""Nodemailer
Contact" <your@email.com>', // sender address
    to: `${req.body.email}`, // list of receivers
    subject: `Email Notification-${req.body.company}`, // Subject
line
    text: `Hi ${req.body.name}! Sending In Email Notification via
nodemailer..`, // plain text body
    html: output, // html body
};
```

```
// send mail with defined transport object
transporter.sendMail(mailOptions, (error, info) => {
  if (error) {
    return console.log(error);
  }
  console.log("Message sent: %s", info.messageId);
  console.log("Preview URL: %s",
nodemailer.getTestMessageUrl(info));

  res.render("contact", { msg: "Email has been sent" });
});
```

```
});
```

```
app.post("/sendmessage", (req, res) => {  
  const accountSid = process.env.TWILIO_ACCOUNT_SID; //  
'AC48fe2e221fc2fe12f04d338113c0b36b'  
  
  const authToken = process.env.TWILIO_AUTH_TOKEN;  
  //'njzfpO1XQIJPBWawY1Ylcvt6UCAqJsg3'  
  
  const client = require("twilio")(accountSid, authToken);  
  
  //var client= new twilio(accountSid)  
  
  client.messages  
    .create({  
      body: `${req.body.messagesms}`,  
      from: "+15005550006",  
      to: `${req.body.phone12}`,  
    })  
  
    .then((message) => console.log("found value in sms",  
message.sid))  
  
    .catch((err) => console.log("error coming from sms", err));  
  
  console.log(`SMS! Sent to ${JSON.stringify(req.body)}`);  
  console.log(`SMS! Sent to 1111 ${JSON.stringify(req.body)}`);  
  res.send(`SMS! Sent to ${req.body.phone12}`);  
}
```

```
});
```

```
app.post("/sendwhatsapp", (req, res) => {  
  //const mJob=schedule.scheduleJob('*/* 2 * * * *',()=>{  
    cron.schedule('* * * * *', () => {  
      wbm.start().then(async () => {  
        const phones = [` ${req.body.phonewhatsapp}`];//to send to  
        multiple phone no seperate by ,  
        const message = ` ${req.body.messagewhatsapp}`;  
        await wbm.send(phones, message);  
        console.log("whatsapp message send to",phones)  
        await wbm.end();  
      }).catch(err => console.log("error in whatsapp",err));  
      console.log(`Whatsapp! Sent to ${JSON.stringify(req.body)}`);  
      res.send(`Whatsapp! Sent to ${req.body.phonewhatsapp}`);  
    });  
    console.log("i ran at the given regular expression",new  
    Date().toString())  
    // mJob.cancel()  
  });
```

```
const port = 6500;  
app.listen(port, () => {  
  console.log(`Server running on ${port}`);  
});
```