

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Project I
Nghiên cứu tạo email server bằng NodeJS giao
thức SMTP

Nguyễn Ngọc Thắng

thang.nn153520@sis.hust.edu.vn

Viện Công nghệ Thông tin và Truyền thông

Giảng viên hướng dẫn: TS. Nguyễn Tuấn Dũng

HÀ NỘI, 1/2021

MỤC LỤC

CHƯƠNG 1: NGHIÊN CỨU NODEJS.....	3
1.1 Tìm hiểu JavaScript.....	3
1.2 Tìm hiểu NodeJs.....	3
1.3 Tìm hiểu NodeJs Modules.....	3
1.3.1 Module trong NodeJs là gì ?.....	3
1.3.2 Cách include một module.....	3
1.4 Tìm hiểu NodeJs Email.....	3
1.4.1 Tìm hiểu Nodemailer Module.....	3
1.4.2 Tìm hiểu cách gửi một email.....	4
1.4.3 Tìm hiểu cách gửi email cho nhiều người nhận.....	4
1.4.4 Tìm hiểu cách gửi văn bản có dạng HTML.....	5
CHƯƠNG 2: NGHIÊN CỨU GIAO THỨC SMTP.....	5
2.1 Giao thức SMTP là gì ?.....	5
2.2 Các thành phần của SMTP.....	6
2.3 Hoạt động của SMTP.....	8
CHƯƠNG 3: THỰC HÀNH TẠO EMAIL SERVER BẰNG NODEJS.....	8
3.1 Tạo thư mục ứng dụng.....	8
3.2 Cài đặt express và nodemailer.....	9
3.3 Tiến hành code.....	9

CHƯƠNG 1:NGHIÊN CỨU NODEJS

1.1 Tìm hiểu Javascript

JavaScript là một ngôn ngữ lập trình thông dịch được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web (phía người dùng) cũng như phía máy chủ (với Nodejs). Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. .js là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.

1.2 Tìm hiểu NodeJs

- Node.js là một môi trường máy chủ mã nguồn mở.
- Node.js cho phép bạn chạy JavaScript trên máy chủ.
- Node.js chạy trên nhiều nền tảng khác nhau (Windows, Linux, Unix, Mac OS X, v.v.)

1.3 Tìm hiểu NodeJs Modules

1.3.1 Module trong NodeJs là gì ?

- Coi các Modules giống như các thư viện JavaScript.
- Một tập hợp các chức năng bạn muốn đưa vào ứng dụng của mình.

1.3.2 Cách include một module

- Để include một module, ta sử dụng hàm `require()` với tên của module:

```
var http = require('http');
```

- Bây giờ ứng dụng của bạn có quyền truy cập vào module HTTP và có thể tạo một máy chủ:

```
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.end('Hello World!');  
}).listen(8080);
```

1.4 Tìm hiểu NodeJs Email

1.4.1 Tìm hiểu Nodemailer Module

- Module Nodemailer giúp bạn dễ dàng gửi email từ máy tính của mình.
- Module Nodemailer có thể được tải và cài đặt bằng npm:

```
C:\Users\Your Name>npm install nodemailer
```

- Sau khi bạn đã tải module Nodemailer, bạn có thể thêm module vào bất kỳ ứng dụng nào:

```
var nodemailer = require('nodemailer');
```

1.4.2 Tìm hiểu cách gửi một email

- Sử dụng tên người dùng và mật khẩu từ nhà cung cấp email đã chọn để gửi email. Sau đây là cách sử dụng tài khoản Gmail để gửi email:

```
var nodemailer = require('nodemailer');
```

```
var transporter = nodemailer.createTransport({
  service: 'gmail',
  auth: {
    user: 'youremail@gmail.com',
    pass: 'yourpassword'
  }
});
```

```
var mailOptions = {
  from: 'youremail@gmail.com',
  to: 'myfriend@yahoo.com',
  subject: 'Sending Email using Node.js',
  text: 'That was easy!'
};
```

```
transporter.sendMail(mailOptions, function(error, info){
  if (error) {
    console.log(error);
  } else {
    console.log('Email sent: ' + info.response);
  }
});
```

1.4.3 Tìm hiểu cách gửi email cho nhiều người nhận

- Để gửi email đến nhiều người nhận, hãy thêm chúng vào thuộc tính "to" của đối tượng mailOptions, được phân viết bằng dấu phẩy:

Ví dụ:

```
var mailOptions = {
  from: 'youremail@gmail.com',
  to: 'myfriend@yahoo.com, myotherfriend@yahoo.com',
```

```
subject: 'Sending Email using Node.js',  
text: 'That was easy!'  
}
```

1.4.4 Tìm hiểu cách gửi văn bản có dạng HTML

- Để gửi văn bản có định dạng HTML trong email, hãy sử dụng thuộc tính "html" thay vì "text"

Ví dụ

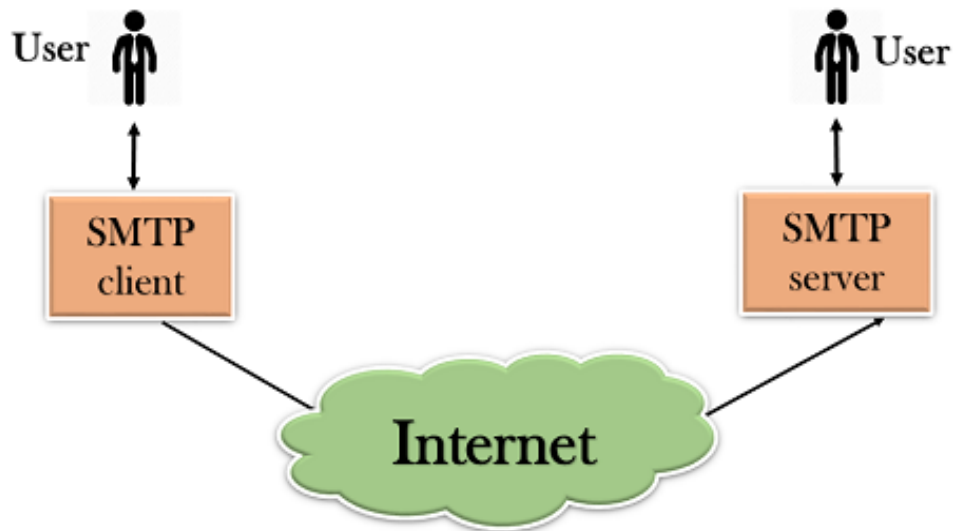
```
var mailOptions = {  
  from: 'youremail@gmail.com',  
  to: 'myfriend@yahoo.com',  
  subject: 'Sending Email using Node.js',  
  html: '<h1>Welcome</h1><p>That was easy!</p>'  
}
```

CHƯƠNG 2: NGHIÊN CỨU GIAO THỨC SMTP

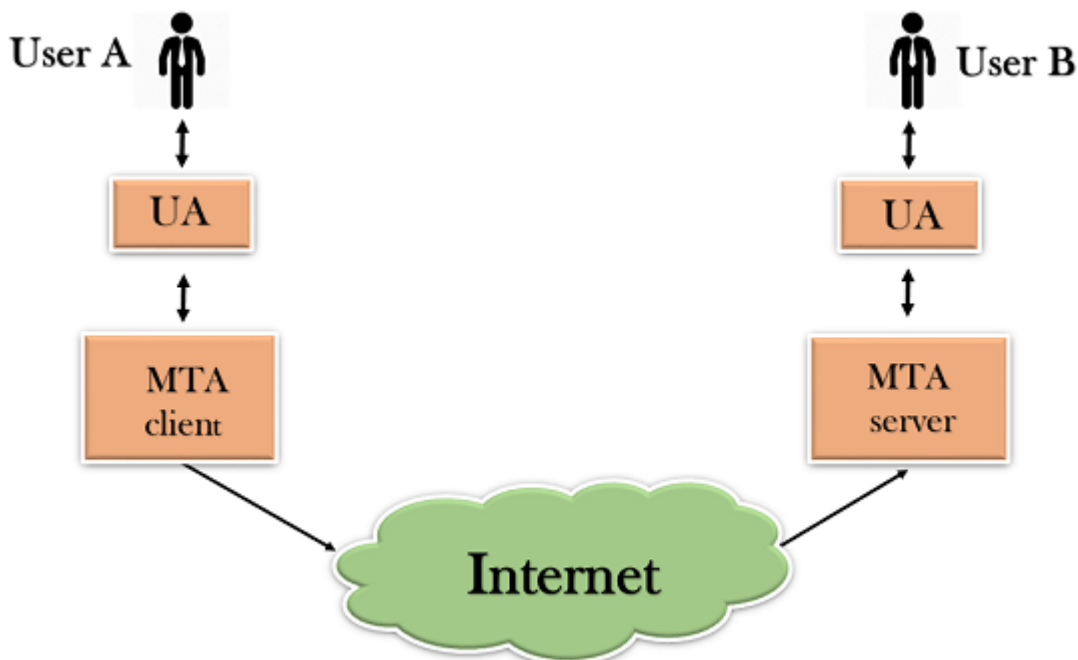
2.1 Giao thức SMTP là gì ?

- SMTP là viết tắt của Simple Mail Transfer Protocol.
- SMTP là một tập hợp các hướng dẫn giao tiếp cho phép phần mềm truyền một thư điện tử qua internet được gọi là Giao thức truyền thư đơn giản.
- Nó là một chương trình được sử dụng để gửi tin nhắn cho người dùng máy tính khác dựa trên địa chỉ e-mail.
- Nó cung cấp trao đổi thư giữa những người dùng trên cùng một máy tính hoặc các máy tính khác nhau
- Mục đích chính của SMTP được sử dụng để thiết lập các quy tắc giao tiếp giữa các máy chủ. Các máy chủ có cách tự nhận dạng và thông báo loại giao tiếp mà chúng đang cố gắng thực hiện. Họ cũng có một cách để xử lý các lỗi như địa chỉ email không chính xác. Ví dụ: nếu địa chỉ người nhận sai, thì máy chủ sẽ nhận được phản hồi kèm theo một thông báo lỗi nào đó.

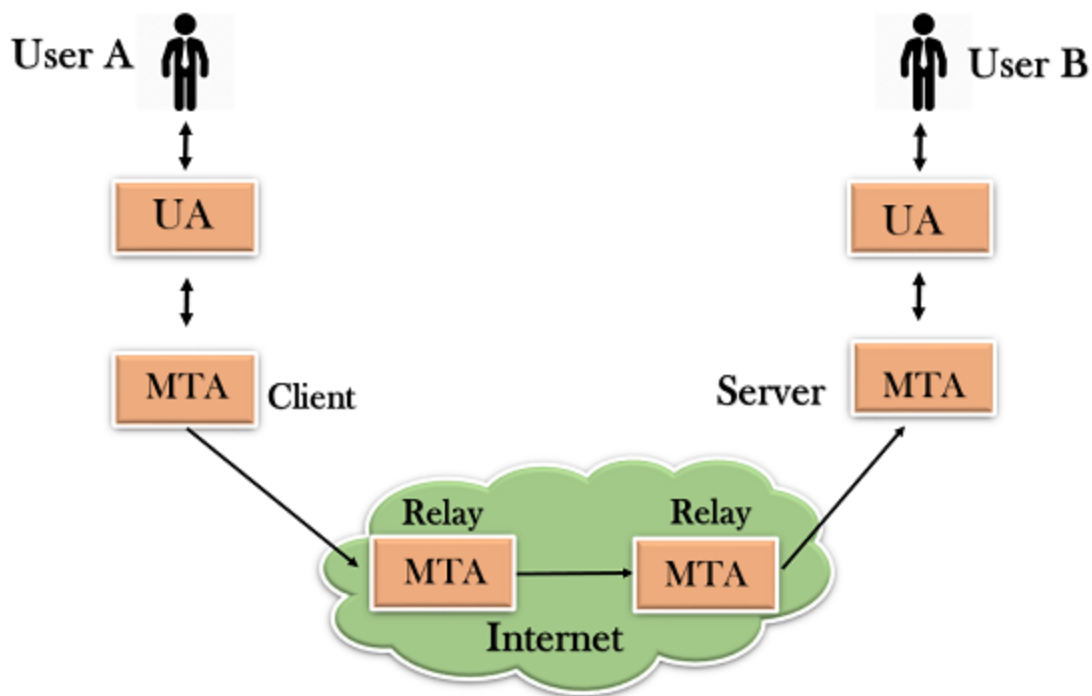
2.2 Các thành phần của SMTP



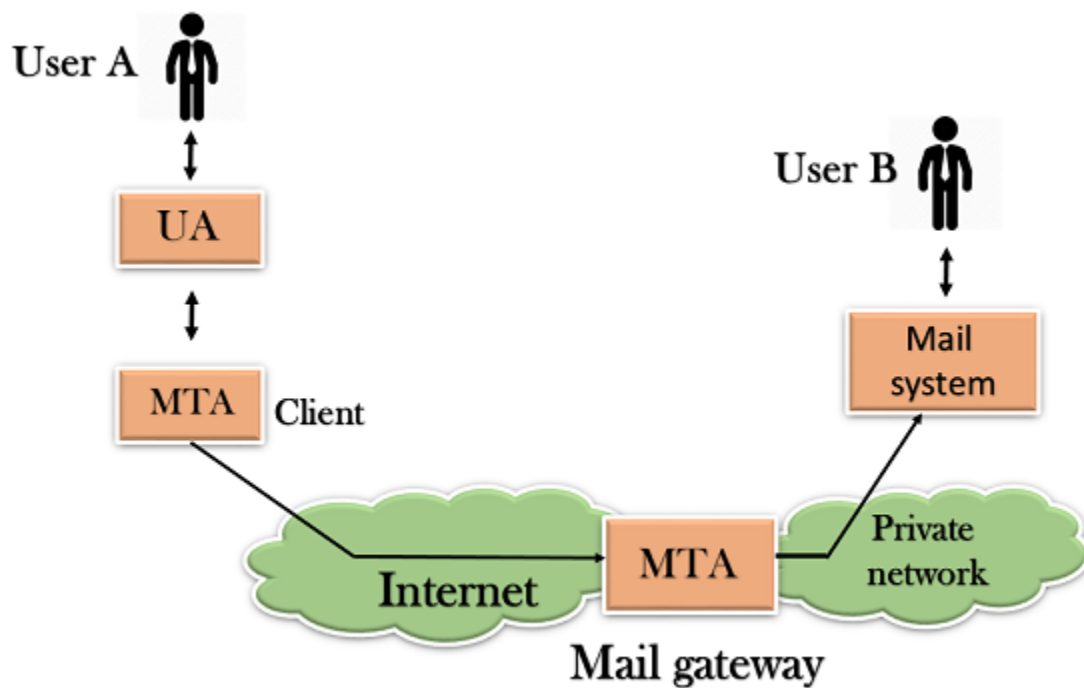
- Đầu tiên, chúng ta sẽ chia máy khách SMTP và máy chủ SMTP thành hai thành phần như tác nhân người dùng (UA) và tác nhân truyền thư (MTA). Tác nhân người dùng (UA) chuẩn bị thư, tạo phong bì và sau đó đặt thư vào phong bì. Đại lý chuyển thư (MTA) chuyển thư này qua internet.



- SMTP cho phép một hệ thống phức tạp hơn bằng cách thêm một hệ thống chuyển tiếp. Thay vì chỉ có một MTA ở phía gửi và một ở phía nhận, có thể thêm nhiều MTA hơn, hoạt động như một máy khách hoặc máy chủ để chuyển tiếp email.



- Hệ thống chuyển tiếp không có giao thức TCP / IP cũng có thể được sử dụng để gửi email đến người dùng và điều này đạt được bằng cách sử dụng cổng thư. Cổng thư là một MTA chuyển tiếp có thể được sử dụng để nhận email.



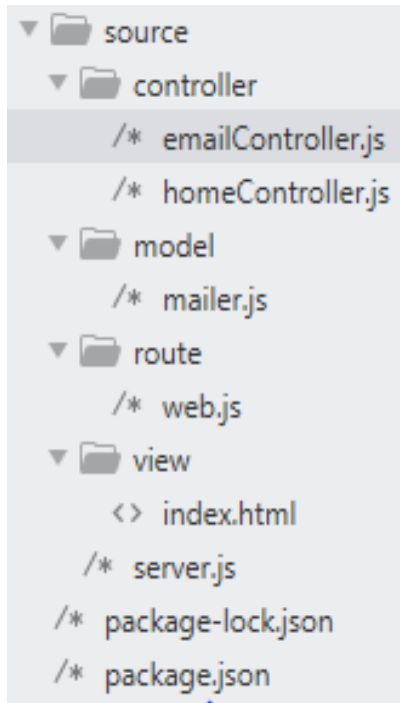
2.3 Hoạt động của SMTP

- **Thành phần của Thư:** Người dùng gửi e-mail bằng cách soạn thư điện tử sử dụng Mail User Agent (MUA). Mail User Agent là một chương trình được sử dụng để gửi và nhận thư. Thư có hai phần: nội dung và tiêu đề. Phần thân là phần chính của thư trong khi phần đầu bao gồm thông tin như địa chỉ người gửi và người nhận. Tiêu đề cũng bao gồm thông tin mô tả chẳng hạn như chủ đề của thư. Trong trường hợp này, nội dung thư giống như một lá thư và tiêu đề giống như một phong bì có chứa địa chỉ của người nhận.
- **Gửi thư:** Sau khi soạn email, ứng dụng thư khách sau đó sẽ gửi e-mail đã hoàn thành đến máy chủ SMTP bằng cách sử dụng SMTP trên cổng TCP 25.
- **Chuyển phát thư:** Địa chỉ email chứa hai phần: tên người dùng của người nhận và tên miền. Ví dụ: vivek@gmail.com, trong đó "vivek" là tên người dùng của người nhận và "gmail.com" là tên miền. Nếu tên miền của địa chỉ email của người nhận khác với tên miền của người gửi, thì MSA sẽ gửi thư đến Mail Transfer Agent (MTA). Để chuyển tiếp email, MTA sẽ tìm miền đích. Nó kiểm tra bản ghi MX từ Hệ thống tên miền để có được miền đích. Bản ghi MX chứa tên miền và địa chỉ IP của miền của người nhận. Khi bản ghi được định vị, MTA kết nối với máy chủ trao đổi để chuyển tiếp thông báo.
- **Nhận và xử lý thư:** Sau khi nhận được thư đến, máy chủ trao đổi sẽ gửi thư đến máy chủ đến (Mail Delivery Agent), nơi lưu trữ e-mail nơi nó chờ người dùng lấy nó.
- **Truy cập và Truy xuất Thư:** Bạn có thể truy xuất email được lưu trữ trong MDA bằng cách sử dụng MUA (Tác nhân Người dùng Thư). MUA có thể được truy cập bằng đăng nhập và mật khẩu.

CHƯƠNG 3: THỰC HÀNH TẠO EMAIL SERVER BẰNG NODEJS

3.1 Tạo thư mục ứng dụng

Tạo thư mục có cấu trúc như sau:



3.2 Cài đặt express và nodemailer

```
npm install --save express nodemailer
```

3.3 Bắt đầu code

1. File **index.html** là form gửi email.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>NodeJS gửi email với nodemailer</title>
</head>
<body>
  <div class="container">
    <h4 class="text-left mt-5">
      NodeJS-Gửi Email
    </h4>
    <hr>
```



```

const sendMail = (to, subject, htmlContent) => {
  // Khởi tạo transporter object sử dụng chuẩn giao thức truyền tải SMTP với các
  // thông tin cấu hình ở trên.

  const transporter = nodeMailer.createTransport({
    host: mailHost,
    port: mailPort,
    secure: false,
    auth: {
      user: adminEmail,
      pass: adminPassword
    }
  })

  const options = {
    from: adminEmail, // địa chỉ admin email bạn dùng để gửi
    to: to, // địa chỉ gửi đến
    subject: subject, // Tiêu đề của mail
    html: htmlContent // Phần nội dung mail
  }

  // hàm transporter.sendMail() này sẽ trả về cho chúng ta một Promise
  return transporter.sendMail(options)
}

module.exports = {
  sendMail: sendMail
}

```

3. File **homeController.js** chứa một function **getHome** sử dụng module **path** để lấy ra file **index.html**. Kết quả là trả về màn hình nhập thông tin email. Mã Nguồn như sau:

```
const path = require('path')

let getHome = (req, res) => {
  return res.sendFile(path.join(`${__dirname}/../view/index.html`))
}
```

```
module.exports = {
  getHome: getHome
}
```

4. File **emailController** sẽ import module **mailer** ,sau đó nhận thông tin request từ client trong **req.body**,cuối cùng gọi hành động gửi email. Mã nguồn như sau:

```
const mailer = require('../model/mailer')

let sendMail = async (req, res) => {
  try {
    // Lấy data truyền lên từ form phía client
    const { to, subject, body } = req.body
    // Thực hiện gửi email
    await mailer.sendMail(to, subject, body)
    // Quá trình gửi email thành công thì gửi về thông báo success cho người dùng
    res.send('<h3>Gửi email thành công.</h3>')
  } catch (error) {
    // Nếu có lỗi thì log ra để kiểm tra và cũng gửi về client
    console.log(error)
    res.send(error)
  }
}

module.exports = {
  sendMail: sendMail}
```

5. File **web.js** để cấu hình router, có 2 router chính:

- Một là router để gọi ra màn hình nhập thông tin email để gửi đi với method là **GET**
- Hai là router để gọi đến **emailController** để thực hiện hành động gửi email với method là **POST**.

Mã nguồn như sau:

```
const express = require('express')

const router = express.Router()

const homeController = require('../controller/homeController')
const emailController = require('../controller/emailController')

let initRoutes = (app) => {

  // Gọi ra trang home
  router.get('/', homeController.getHome)

  // Gọi hành động gửi email
  router.post('/send-email', emailController.sendMail)

  return app.use('/', router)

};

module.exports = initRoutes
```

6. File **server.js** để chạy ứng dụng **Node.js**

```
const express = require('express')

const app = express()

const initRoutes = require('./route/web')

// Cho phép lý dữ liệu từ form method POST
app.use(express.urlencoded({extended: true}))

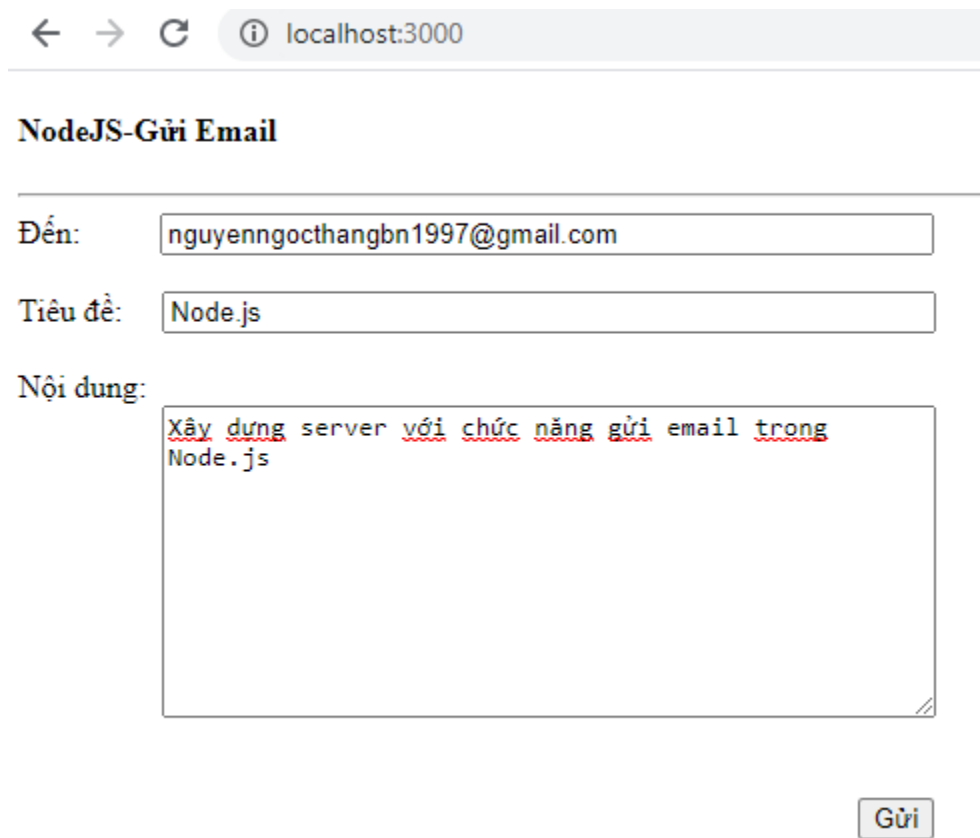
// Khởi tạo các routes cho ứng dụng
initRoutes(app)

// chọn một port để chạy ứng dụng tại local
```

```
const port = 3000
app.listen(port, () => {
  console.log(`Bat Server thanh cong, Server running at localhost:${port}/`)
})
```

Kết quả

Form nhập thông tin email gửi đi như sau:



The screenshot shows a web browser window with the address bar displaying 'localhost:3000'. The page title is 'NodeJS-Gửi Email'. Below the title, there is a form with three fields: 'Đến:' (To) with the value 'nguyenngocthangbn1997@gmail.com', 'Tiêu đề:' (Subject) with the value 'Node.js', and 'Nội dung:' (Content) with the value 'Xây dựng server với chức năng gửi email trong Node.js'. A 'Gửi' (Send) button is located at the bottom right of the form.

← → ↻ ⓘ localhost:3000

NodeJS-Gửi Email

Đến:

Tiêu đề:

Nội dung:

Email gửi thành công:

