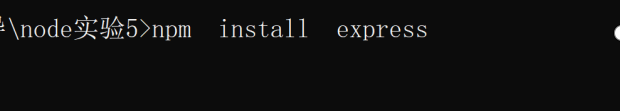
**实验五 express模块构建web服务器程序**

实验内容

1. 实验准备-下载express模块

在磁盘中建立一个文件夹如：node实验五，进入命令行窗口下载express模块，如图



在vscode中“文件”菜单选打开文件夹，选择打开上述文件夹，本次实验文件都在该文件夹中存放

1. 应用express模块构建一个简单的web服务器程序

var express=require("express")

var app=express();

app.get("/",function(req,res){

console.log("接收到请求")

// res.end("这是响应的内容")

res.send("这是响应的内容")

})

app.get("/login",function(req,res){

res.send("这是登录的响应")

})

app.listen(3000,function(){

console.log("server started")

})

在命令行窗口执行启动服务器

然后在浏览器地址栏分别输入以下地址后回车，在浏览器窗口查看结果：

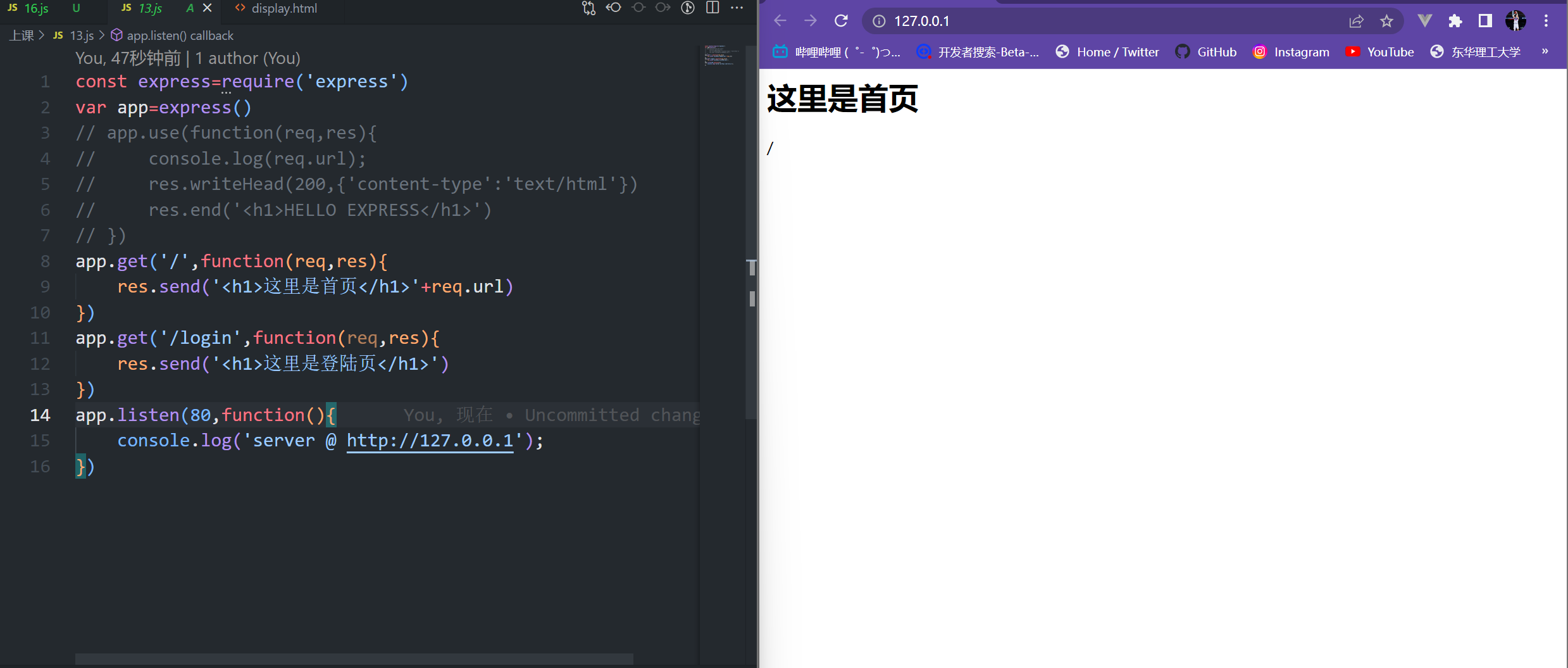
[**http://127.0.0.1:3000**](http://127.0.0.1:3000)**/**

[**http://127.0.0.1:3000/login**](http://127.0.0.1:3000/login)

1. 要响应如下URL请求，应如何处理

[**http://127.0.0.1:3000**](http://127.0.0.1:3000)**/register**

[**http://127.0.0.1:3000/**](http://127.0.0.1:3000/)**findbackpassword**



1. Express处理get 以及post请求
2. 在实验目录下新建一个views文件夹，并在该文件夹下新建一个login.html

<form action="/login" method="post">

请输入姓名：<input type="text" name=”username><br>

请输入密码：<input type="password" name=”pwd><br>

<input type="submit" value="登录" >

</form>

2) 在实验目录下新建一个 express处理get以及post请求.js文件,如下

var express=require("express")

var fs=require("fs")

var app=express()

app.get("/login",function(req,res){

fs.readFile("./views/login.html","utf8",function(err,data){

//console.log(data)

res.send(data)

})

})

app.post("/login",function(req,res){

res.send("这是login post 的响应")

})

app.listen(3000,function(){

console.log("server started")

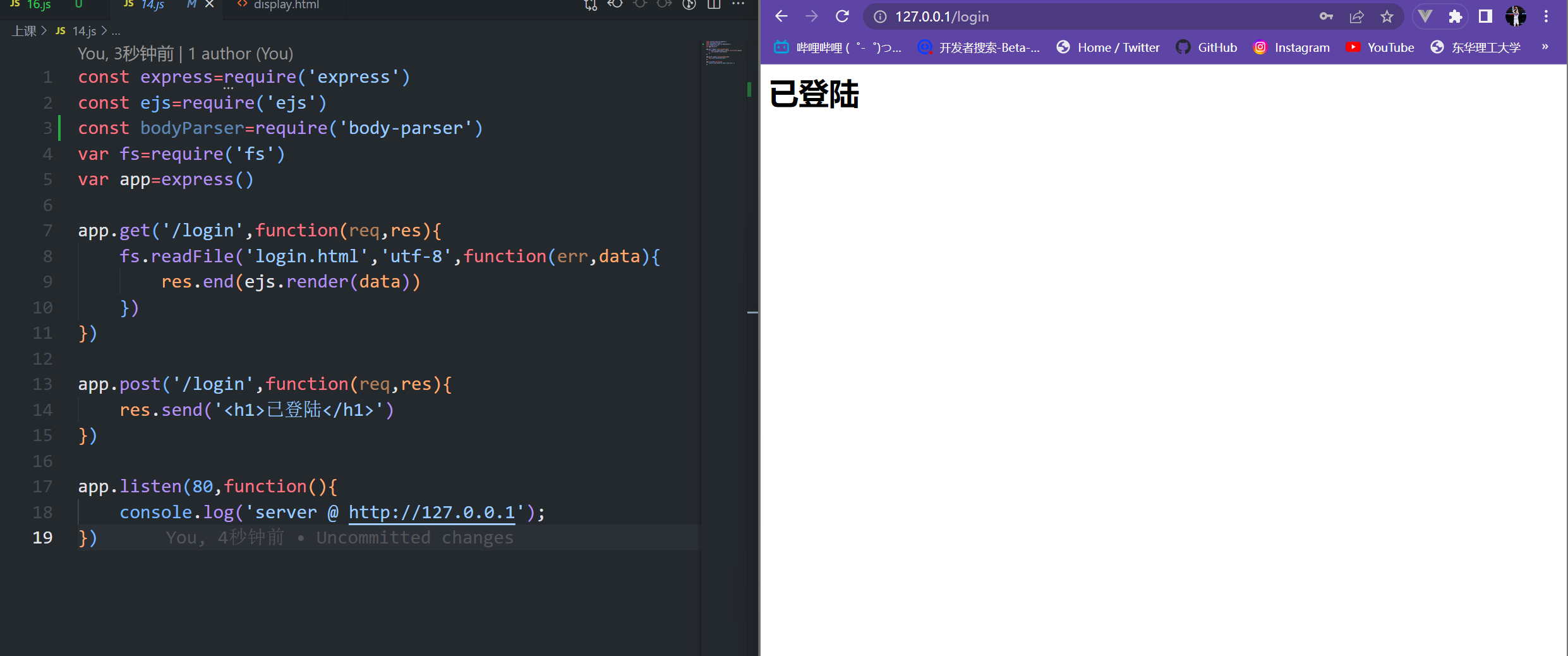
})

在命令行窗口执行启动服务器

然后在浏览器地址栏分别输入以下地址后回车，在浏览器窗口查看结果：

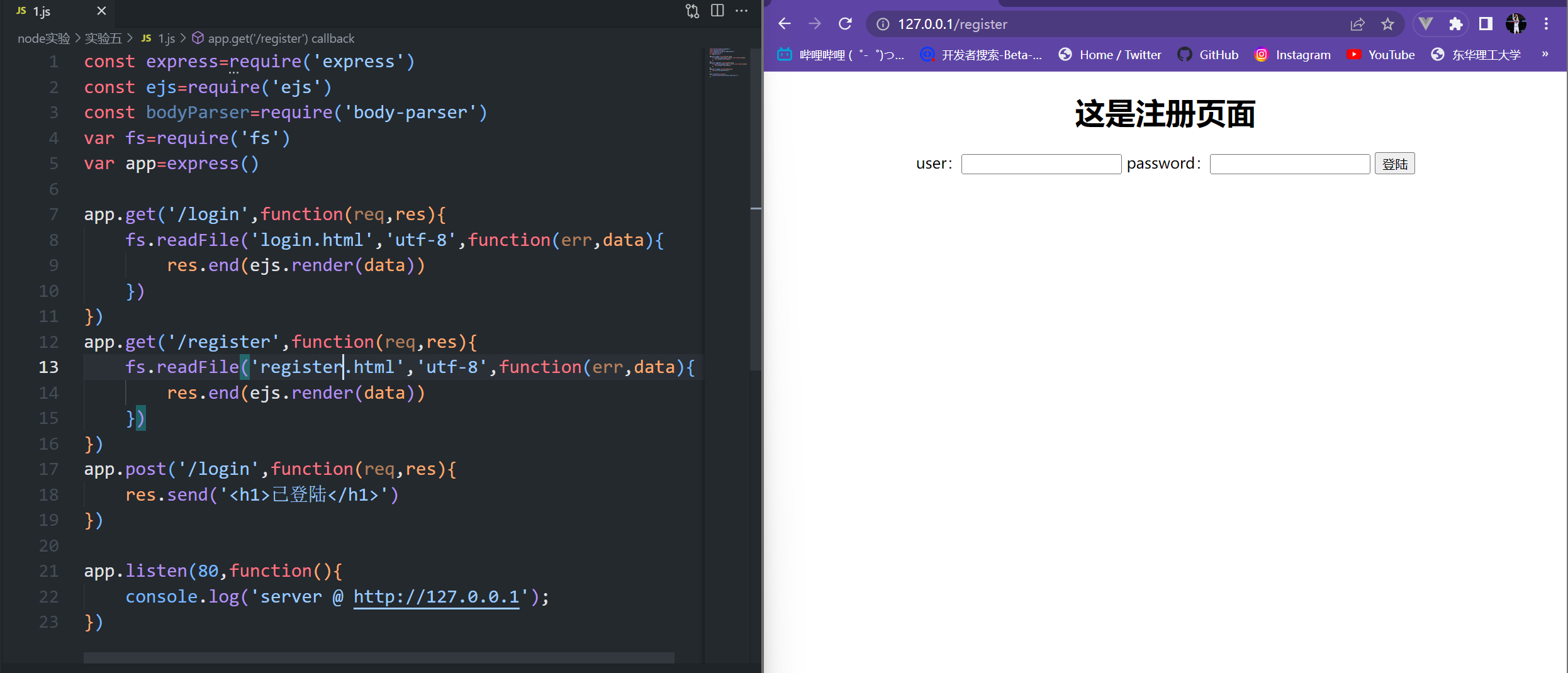
[**http://127.0.0.1:3000/login**](http://127.0.0.1:3000/login)

点击登录按钮，看有什么效果



1. 思考：建立一个register.html文件，修改上述程序，在浏览器地址栏中输入**<http://127.0.0.1:3000/register>**

**能在浏览器中浏览该注册页面，当点击提交按钮时，能有响应**



1. 通过body parser得到post提交的数据
2. 先下载第三方模块body-parser

Npm install body-parser

1. 在第3题的基础上做一定的修改

var express=require("express")

var bodyParser=require("body-parser")

var fs=require("fs")

var app=express()

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

app.listen(3200,function(){

console.log("server start ")

})

app.get("/login",function(req,res){

fs.readFile("./views/login.html","utf8",function(err,data){

res.send(data)

})

})

app.post("/login",function(req,res){

console.log(req.body)

res.send(req.body.username)

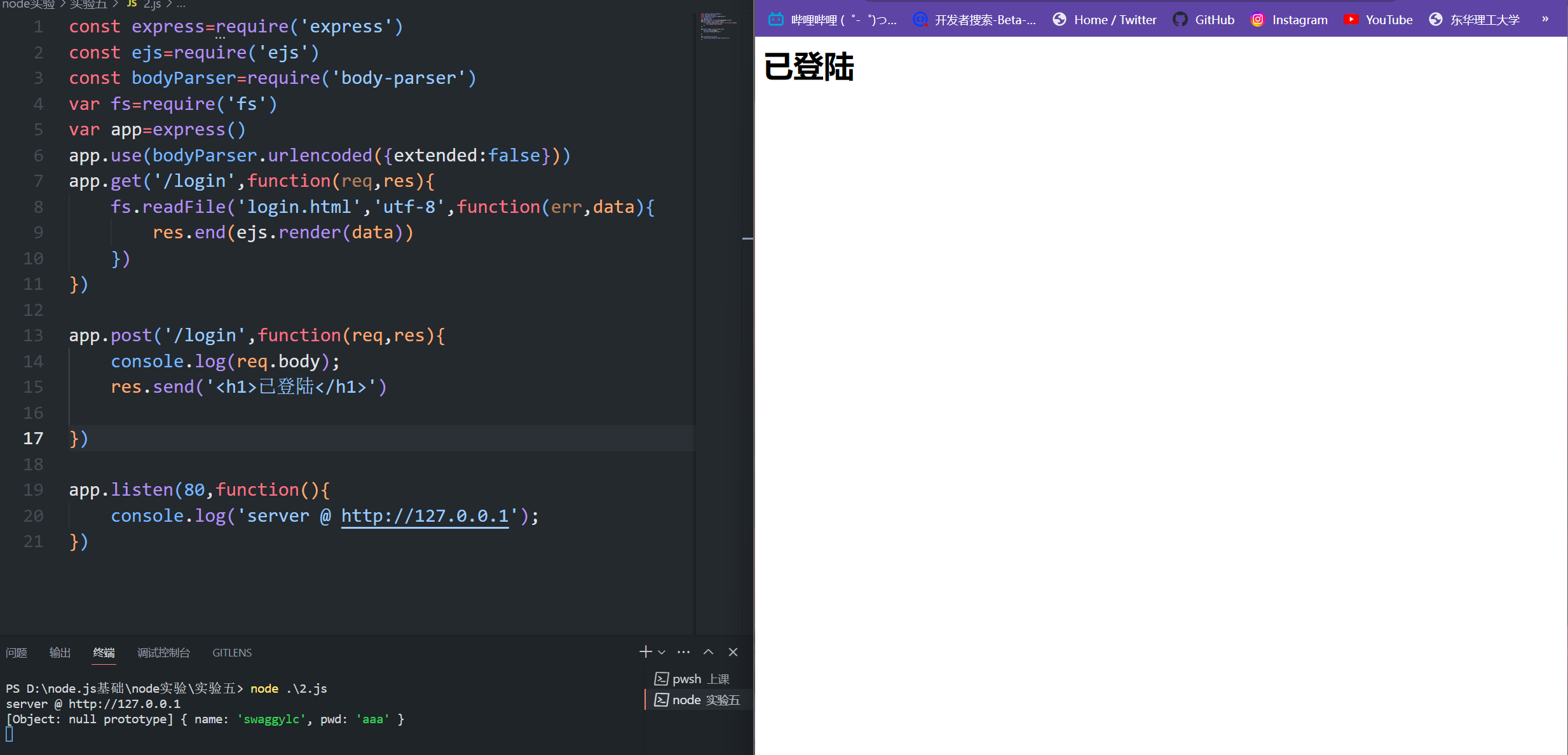
})

在命令行窗口执行启动服务器

然后在浏览器地址栏分别输入以下地址后回车，在浏览器窗口查看结果：

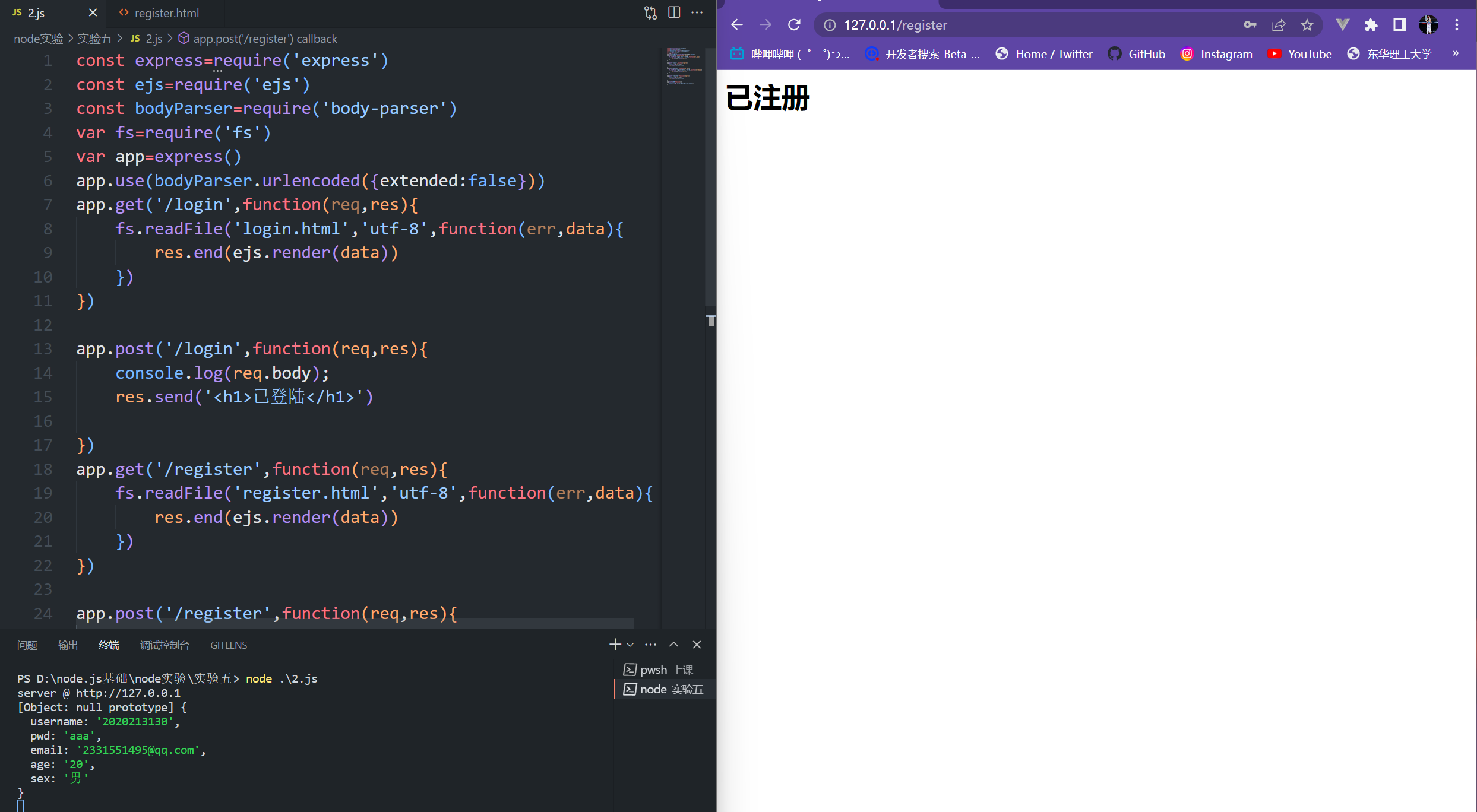
[**http://127.0.0.1:3000/login**](http://127.0.0.1:3000/login)

点击登录按钮，看有什么效果

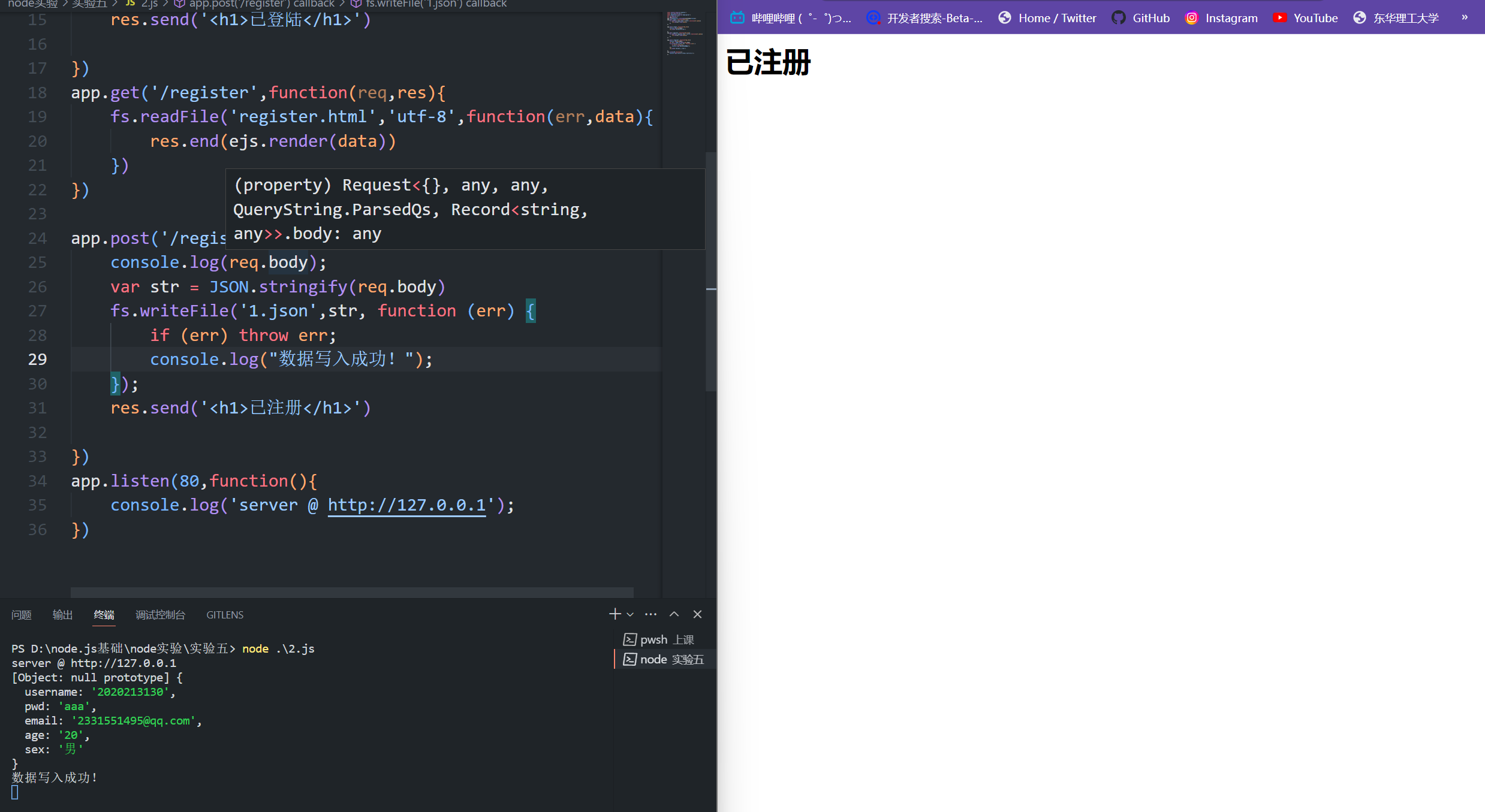


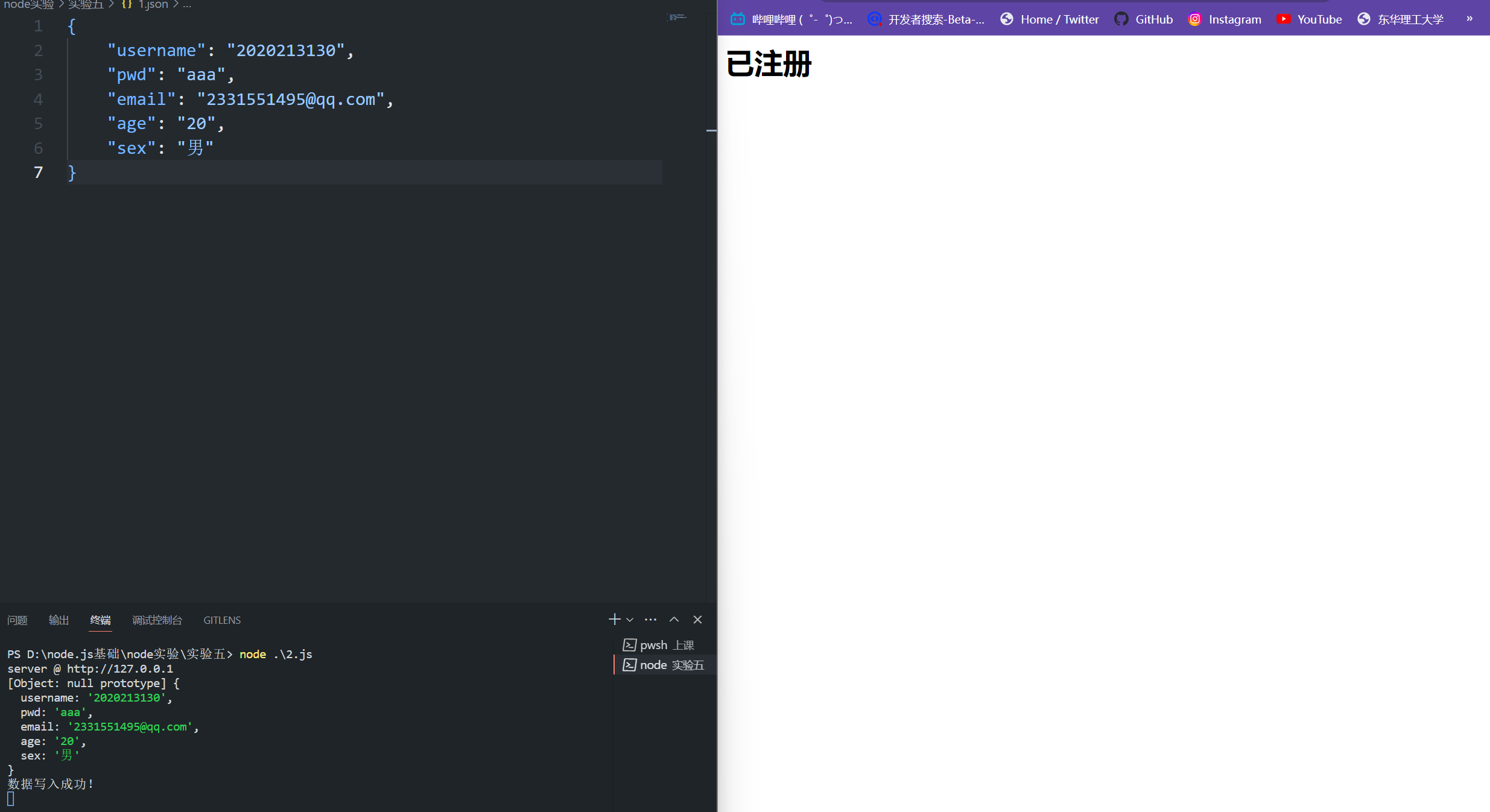
1. 思考，修改上述程序，在浏览器地址栏中输入**<http://127.0.0.1:3000/register>**

**能在浏览器中浏览该注册页面，当点击提交按钮时，能得到注册页面输入的数据（5项数据）并输出**

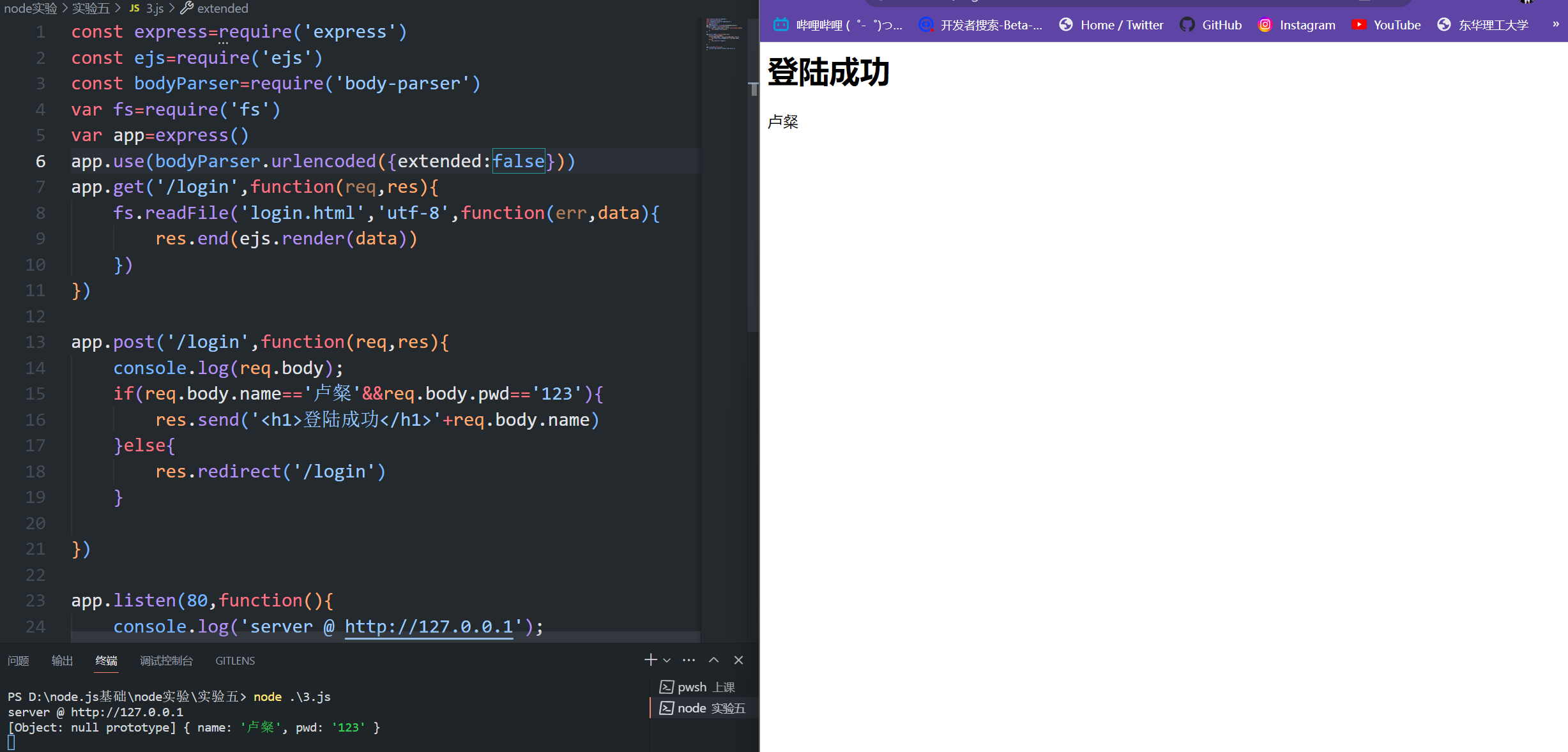


1. **能否结合实验二将注册信息写入一个 JSON文件中去**





1. **编写程序，实现一个功能：先调出登录页，输入姓名密码后，判断，如姓名是你的名字、密码为123456，则跳转到登录成功页，页面上显示姓名，否则跳转到登录页**



1. **收获与心得**

本次实验我们了解了express模块，学会了使用express创建一个基本的服务器，和使用express处理get和post请求，也能够使用body-parser第三方模块得到表单提交的数据，也能够结合实验二将注册的数据写入json文件中，并通过简单的判断是否重新跳转至登陆页，本次的实验很有意义。