

CI 103 Introduction to Computing and Information Security

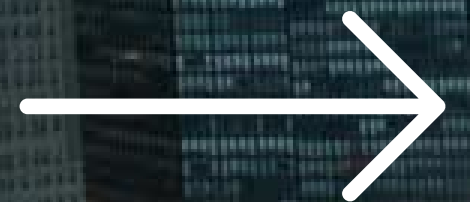
Release Presentation for



ECONOMY FUELER

CONTENT

- 01** Team Member Work Allocation
- 02** Achieved Functions & Functions Flowcharts
- 03** Product Demonstration
- 04** Technology Sharing 4: Using Django to send emails to the users to enhance the register system
- 05** Technology Sharing 5: HTML `<input type="file">` Problems
- 06** Technology Sharing 6: CSRF Token
- 07** Technology Sharing 7: Import Database



01

Team Member Work Allocation

The bottom half of the slide features a series of thin, light blue wavy lines that flow horizontally across the frame, creating a sense of movement and depth.



ECONOMY FUELER

Lixiao Yang

Role: Project Manager

02

- Integrated scheduling and planning
- Data ETL process
- Refining models through financial material
- Assist in frontend development
- Assist in database design & development



ECONOMY FUELER

Jiahe Xie

02

Role: Technical Director & Backend Developer

- Overall backend development including the interactions with frontend and the database
- Algorithms development
- Backend management system development



ECONOMY FUELER

Aoyu Liu

Role: Frontend & Data Visualization Developer

- Frontend development
- Developing the login and register system
- Data visualization development
- Assist in the development of the interactions among frontend, backend and database



ECONOMY FUELER

Xinyue Zeng

Role: Database Designer & Data Processor

02

- Database design
- Create relational database
- Assist in data ETL process



ECONOMY FUELER

Huiru Chang

Role: Data Processor & Finance Modeler

02

- Refining models through financial material
- Data ETL process
- Assist in the development of algorithms



02

Achieved Functions & Functions Flowchart

Achieved Functions (1): Register, Authentication and Log in

User:

- Register new user accounts
- Mailbox verification
- Log in using account and password

Administrator:

- Save user information into database

欢迎注册

用户名:

密码:

确认密码:

邮箱地址:

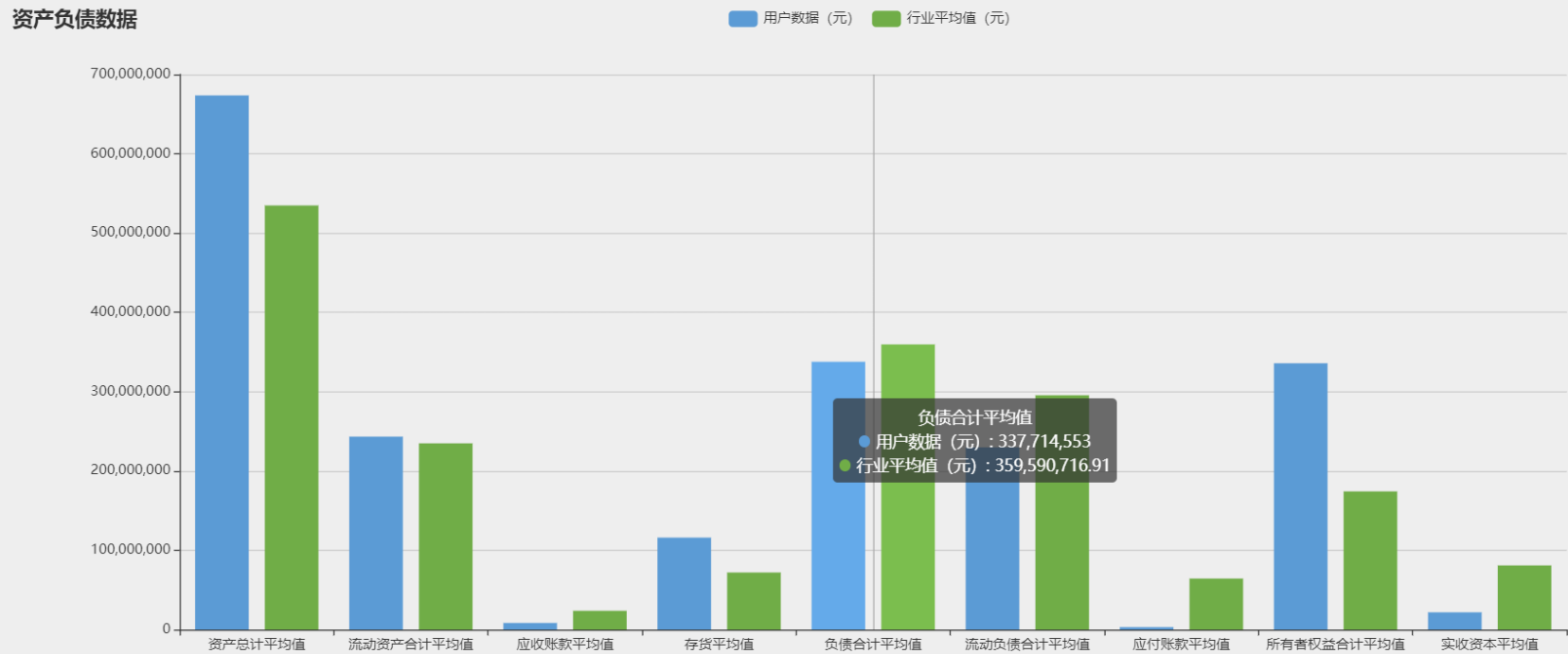
性别:

验证码: 

Achieved Functions (2): Simplified Analysis

- Read the financial sheet files (1 year)
- Display visual charts (Compared with the average of the industry)
- Detailed analysis of the original data

资产负债数据

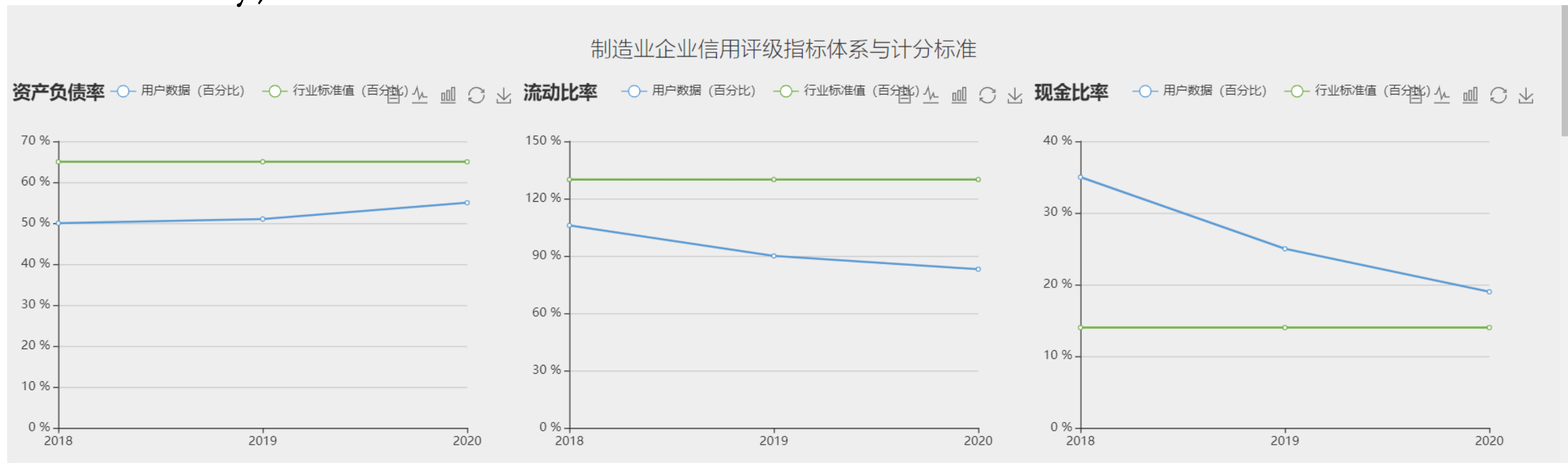


对比结果阐述

指标名称	数值大小	标准值	分析结果
流动比率	1.06	1.3	低于正常值，企业的短期偿债风险较大
速动比率	0.55	1	低于1的速动比率通常被认爲是短期偿债能力偏低
存货周期率	7.75	3	
存货周转天数	46.0	120	存货周转天数越少,表明存货周转次数越多,平均存货越少
应收账款周转率	72.77	3	应收账款周转率越高，说明其收回越快。
应收账款周转天数	4.0	100	营运资金过多呆滞在应收账款上，影响正常资金周转及偿债能力。
经营周期	50.0	200	营运资金过多呆滞在应收账款上，影响正常资金周转及偿债能力。
流动资产周转率	2.29	1	
总资产周转率	0.81	0.7	达到85%及以上时，应视爲发出预警信号，企业应提起足够的注意。
资产负债率	0.5	0.65	达到85%及以上时，应视爲发出预警信号，企业应提起足够的注意。
产权比率	1.01	1.2	产权比率低，是低风险、低报酬的财务结构
有形净值债务率	1.35	1.5	较低的比率说明企业有良好的偿债能力，举债规模正常。
已获利息倍数	1.05	2.5	已获利息倍数越低，企业长期偿债能力越低。
销售净利率	0.05	0.1	销售净利率下降说明企业销售的盈利能力降低
销售毛利率	0.25	0.15	销售毛利率高就表示企业生产或者采购成本低，产品的附加值高，产品可能比较先进，市场需求量高。表示缴

Achieved Functions (3): Loan Feasibility Analysis

- Read the financial sheet files (3 years)
- Display visual charts (Display the changing trends & Compared with the average of the industry)



Achieved Functions (3): Loan Feasibility Analysis

- Detailed text analysis of certain financial indexes
- Loan possibility score & rating
- Text explanation of the rating

信用等级阐述

A

信用较好

债务没有问题。该类企业具有良好的信用记录，经营处于良性循环状态，但是可能存在一些影响其未来经营与发展的不确定因素。

财务数据指标分析			
指标名称	指标值	分数	分析
实际到位的注册资本		4	
治理机制		4	
管理水平		4	
领导者素质		4	
市场前景、发展规划 和实施条件		2	
财务信息质量指标		-10	
资产负债率	0.5	10.0	一方面,说明企业资金充足,不需要借债经营; 另一方面,说明企业举债经营能力不足。
流动比率	1.06	4.99	低于正常值，企业的短期偿债风险较大
现金比率	0.35	6.0	比率过高，就意味着企业流动负债未能得到合理运用，而现金类资产获利能力低，这类资产金额太高会导致企业机会成本增加
经营活动现金流负债比	0.16	4.0	这一比率越高，企业承担债务的能力越强，破产的可能性越小。
净现金流量	净现金流量	2	高：经营效益、财产状况好。过高：可能是资金闲置，将影响企业资金能力的发挥
主营业务利润率	0.0	0.0	注册资本低的劣势是不容易获得商业伙伴信任及从事业务会受到限制。一个公司的注册资金在一定程度上代表其实力和信誉，如果注册资本过低，很多业务无法参加。
资本回报率	0.07	4.0	公司治理机制不完善,将极大地制约企业进一步发展壮大,甚至导致破产

Achieved Functions (4): Backend Management System

- Display user information
- Search database through management system
- Display user information (including the user names and encrypted passwords)

EcoFuler后台管理

消息 邮件 

首页

用户管理

数据库管理

首页 / 数据库管理

各行业贷款数据

南部地区省份

全国数据

北方地区省份

Q Search Information 搜索

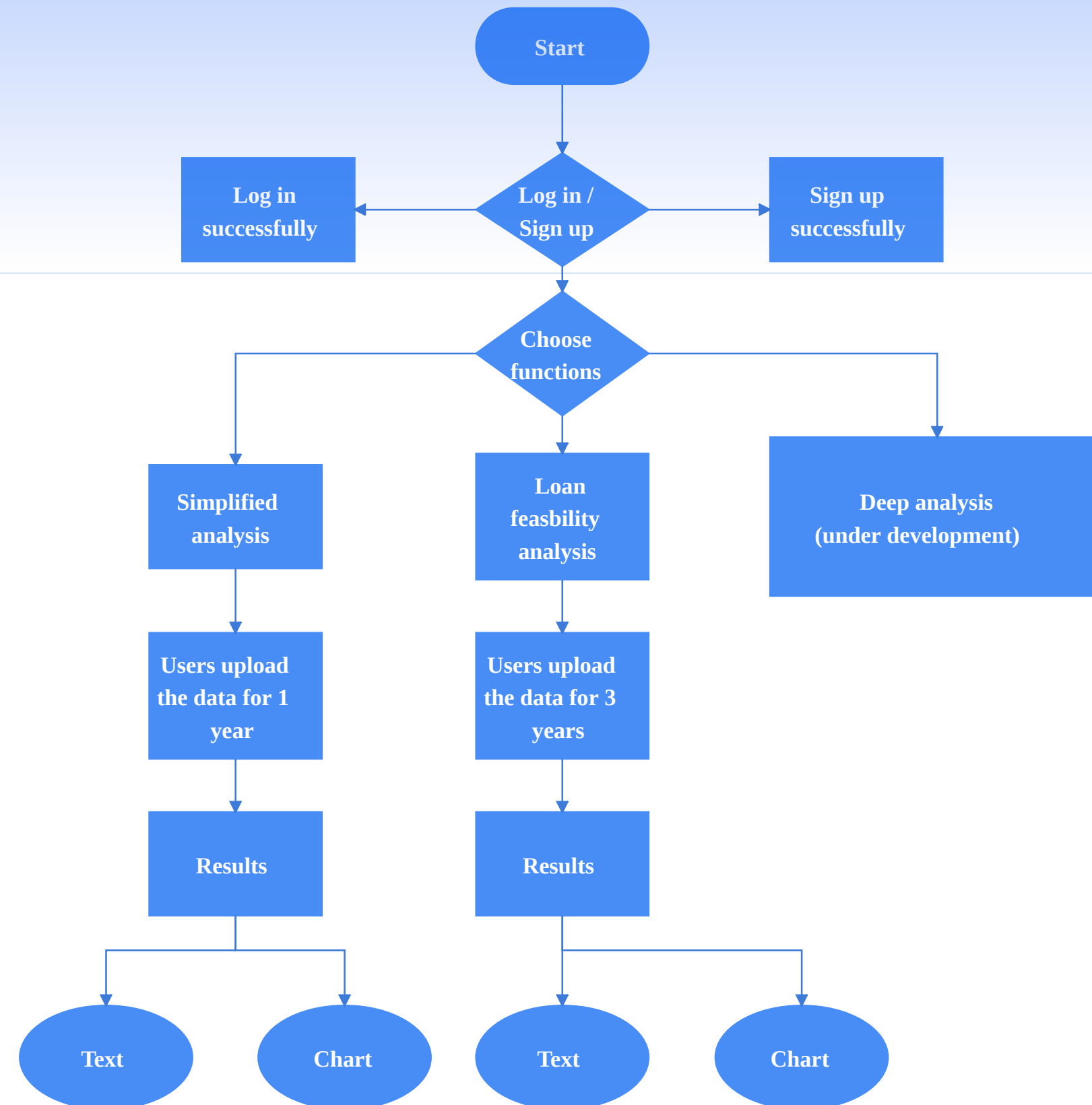
+批量增加

批量删除

id	地区	时间年份	行业名称	指标名称	指标值	操作
1	全国	2012	农副食品加工业	资产总计平均值	104911969.94	删除 编辑
10	全国	2012	化学原料和化学制品制造业	资产总计平均值	225297923.52	删除 编辑
100	全国	2012	专用设备制造业	存货平均值	28798778.869	删除 编辑
101	全国	2012	汽车制造业	存货平均值	38878349.601	删除 编辑
102	全国	2012	铁路、船舶、航空航天和其他运输设备制造业	存货平均值	66670027.305	删除 编辑
103	全国	2012	电气机械和器材制造业	存货平均值	27017763.002	删除 编辑
104	全国	2012	计算机、通信和其他电子设备制造业	存货平均值	52198896.82	删除 编辑
105	全国	2012	仪器仪表制造业	存货平均值	24180168.332	删除 编辑
106	全国	2012	其他制造业	存货平均值	16574002.574	删除 编辑

用户id	用户姓名	用户密码
4	test2	1b373fb10a017bc7386a26c8379f998d897ccbc6a63fc6253ef1cb51d409ba5d
2	test	1b373fb10a017bc7386a26c8379f998d897ccbc6a63fc6253ef1cb51d409ba5d

Function Flowchart



03

Product Demonstration



Technology Sharing 4: Using Django to send emails to the users to enhance the register system

Using Django to send emails to the users to enhance the register system

The module which is set by Django called Django.core.mail can achieve the function of sending emails. First, we choose the QQ postbox as the SMTP server, and we start the POP3/SMTP service manually.

POP3/IMAP/SMTP/Exchange/CardDAV/CalDAV服务

开启服务：	POP3/SMTP服务 (如何使用 Foxmail 等软件收发邮件?)	已开启 关闭
	IMAP/SMTP服务 (什么是 IMAP, 它又是如何设置?)	已关闭 开启
	Exchange服务 (什么是Exchange, 它又是如何设置?)	已关闭 开启
	CardDAV/CalDAV服务 (什么是CardDAV/CalDAV, 它又是如何设置?)	已关闭 开启
	(POP3/IMAP/SMTP/CardDAV/CalDAV服务均支持SSL连接。如何设置?)	

Then, we need to configure some information in the settings.py file.

```
EMAIL_USE_SSL = True
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.qq.com'
EMAIL_PORT = 465
EMAIL_HOST_USER = '1411547623@qq.com'
EMAIL_HOST_PASSWORD = 'lvuxkdkpepzhihha'
DEFAULT_FROM_EMAIL = EMAIL_HOST_USER
CONFIRM_DAYS = 7
```

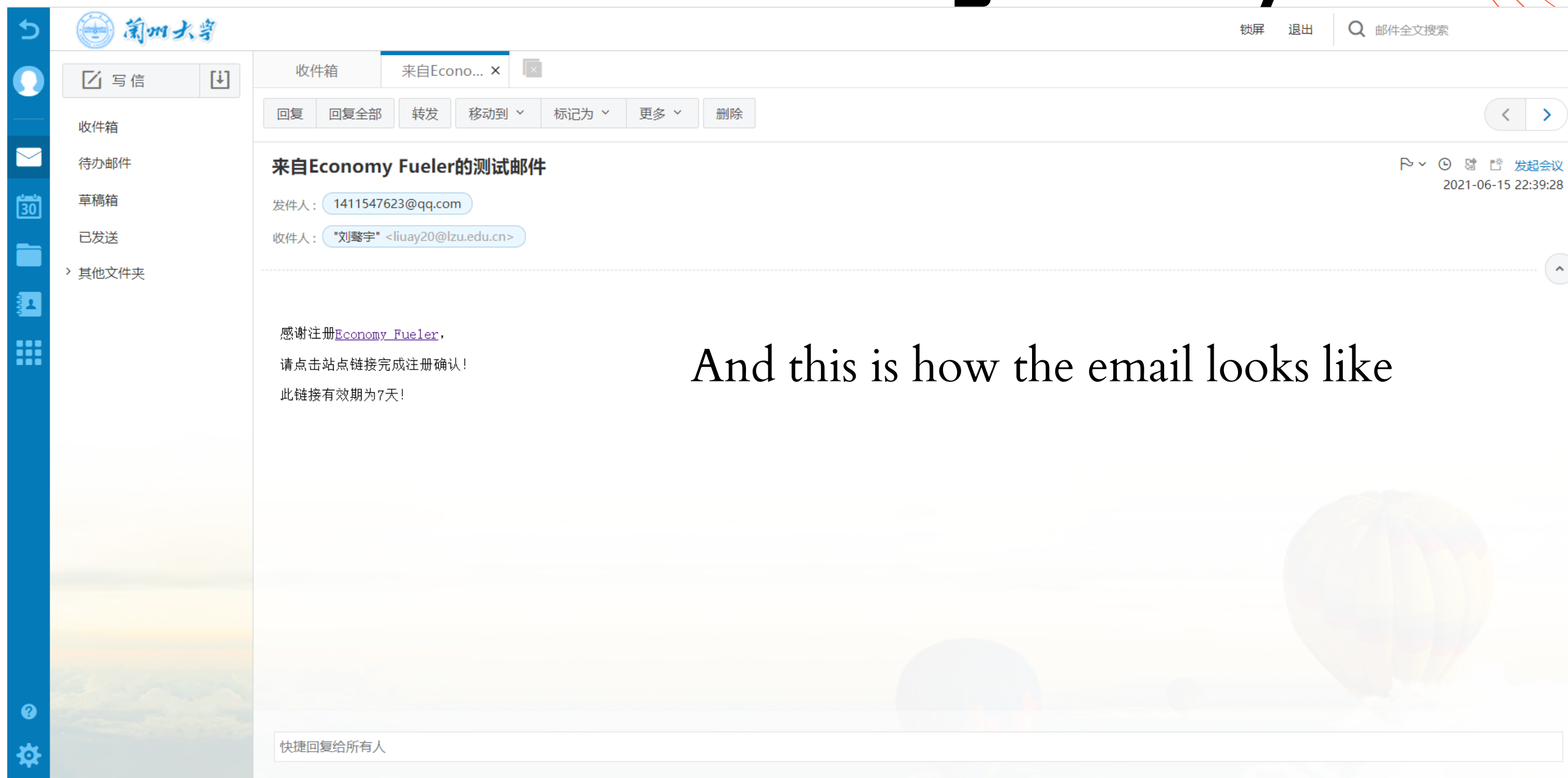
This is how we define the send_email function in the views.py file.

```
def send_email(email, code):
    from django.core.mail import EmailMultiAlternatives

    subject, from_email, to = '来自Economy Fueler的测试邮件', '1411547623@qq.com', email
    text_content = '欢迎使用Economy Fueler!'
    html_content = '''
        <p>感谢注册<a href="http://{}/confirm?code={}" target=blank>Economy Fueler</a>, \
        <p>请点击站点链接完成注册确认! </p>
        <p>此链接有效期为{}天! </p>
        '''.format('127.0.0.1:8000', code, settings.CONFIRM_DAYS)
    msg = EmailMultiAlternatives(subject, text_content, from_email, [to])
    msg.attach_alternative(html_content, "text/html")
    msg.send()
    print('done')
```

Using Django to send emails to the users to enhance the register system

04



And this is how the email looks like

Using Django to send emails to the users to enhance the register system

04

Once you clicked the link, you will be linked to the login page, for this is defined in the views.py file, and we also defined several warnings if the users operate improperly:

```
send_email(email, code)
```

```
message = '请前往注册邮箱, 进行邮件确认! '
```

```
return render(request, 'login/confirm.html', locals())
```

```
if not user.has_confirmed:
```

```
    message = "该用户还未通过邮件确认! "
```

```
    return render(request, 'login/login.html', locals())
```

```
if same_email_user:
```

```
    message = '该邮箱地址已被注册, 请使用别的邮箱! '
```

```
    return render(request, 'login/register.html', locals())
```

```
def user_confirm(request):
    code = request.GET.get('code', None)
    message = ''
    try:
        confirm = models.ConfirmString.objects.get(code=code)
    except:
        message = '无效的确认请求!'
        return render(request, 'login/confirm.html', locals())
```

```
c_time = confirm.c_time.replace(tzinfo=pytz.UTC)
```

```
now = datetime.datetime.now().replace(tzinfo=pytz.UTC)
```

```
if now > c_time + datetime.timedelta(settings.CONFIRM_DAYS):
```

```
    confirm.user.delete()
```

```
    message = '您的邮件已经过期! 请重新注册! '
```

```
    return render(request, 'login/confirm.html', locals())
```

```
else:
```

```
    confirm.user.has_confirmed = True
```

```
    confirm.user.save()
```

```
    confirm.delete()
```

```
    message = '感谢确认, 请使用账户登录! '
```

```
    return render(request, 'login/confirm.html', locals())
```

Using Django to send emails to the users to enhance the register system

04

In the admin page, we can see the user's information, including the user's email address and whether the email address has confirmed:

Django administration

WELCOME, **ECONOMY_FUELER**. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home › Login › 用户 › test2

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

Users [+ Add](#)

LOGIN

用户 [+ Add](#)

确认码 [+ Add](#)

Change 用户

test2

HISTORY

Name:

test2

Password:

faa32c9525f6ee1b1ecdc1bcba55fac894230f

Email:

liuay20@lzu.edu.cn

Sex:

男

☒ Has confirmed

Delete

Save and add another

Save and continue editing

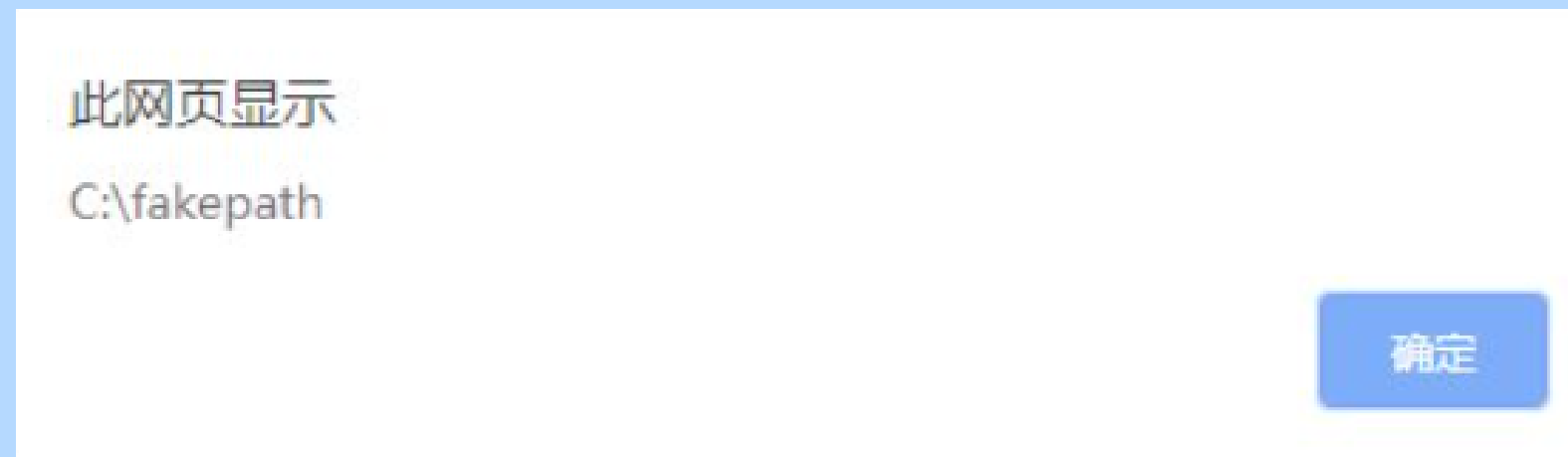
SAVE

05

**Technology Sharing 5:
HTML `<input type="file">`
Problems**

HTML `<input type="file">` problems

PROBLEM: The browser will recognize this tag by the file path on the computer Instead of the file itself. But the path is not the truth absolute path.



Result: The fakepath message can't be recognized by backend

HTML <input type="file"> problems

SOLUTION:

The data in the excel that users update should be processed by JavaScript.

```
readXlsxFile(file_add1.files[0]).then(function (data) {  
    data = JSON.stringify(data)  
    console.log(data)  
    $.ajax({  
        url: "/model_add/",  
        type: "POST",  
        data: {"add_data1": data},  
        success: function (args) {  
            console.log(args)  
        }  
    })  
})  
}
```



06

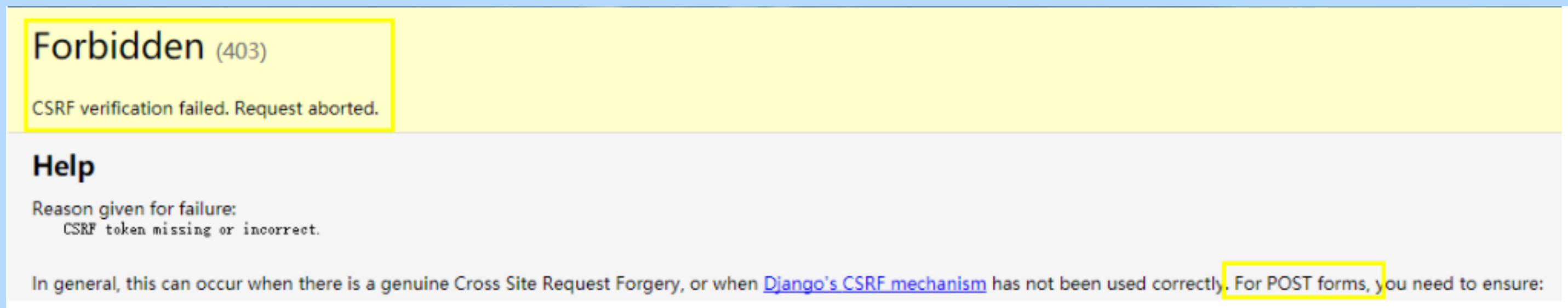
Technology Sharing 6: CRSF Token

CSRF Token

PROBLEM:

Django server has the csrf token mechanism to protect the server from csrf attack.

So if we send post to the server. It will check automatically and determine whether let visitor come. (including ajax). It will cause the problem.



CSRF Token

SOLUTION:

Write the `@csrf_exempt` on the view function, it will allow the post request

```
s_login.py ×  urls.py ×  models.py ×  settings.py ×  user_manage.html
]
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    # 'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

from django.views.decorators.csrf import csrf_exempt, csrf_protect

import test2
# Create your views here.
@csrf_exempt
def edit(request, a1):
    print(a1)

    return HttpResponse('...')
```


07

Technology Sharing 7: Import Database

Import Database

PROBLEM:

Download the Navicat to run the *.sql to get the database from other computer

Use the following orders in the terminal

```
python manage.py inspectdb > filename.py
```

```
python manage.py makemigrations
```

```
python manage.py migrate
```

Cause an app has two models (means an app correspond 2 database).

After reload the Django project, we can migrate the data about the import database.

SOLUTION:

Create a new app and add data by the model in this model and make migrations and migrate again.

The background features a dark red, almost black, solid color. Overlaid on this are numerous thin, wavy, orange lines that flow across the frame, creating a sense of movement and depth. These lines are concentrated in the upper and lower portions, framing the central text.

Thank You for Listening