

研究報告

中市新太
富山県立大学 情報システム工学科
u020025@st.pu-toyama.ac.jp

January 9, 2024

Login

ユーザー名:

パスワード:

ログイン

Registration

新しいユーザー名:

新しいパスワード:

名前:

新規登録

```

===== ログイン =====
@app.route('/login', methods=['POST'])
def login():
    username = request.form.get('username')
    password = request.form.get('password')

    if not username or not password:
        return render_template('login.html', login_message='ユーザー名とパスワードを入力してください')

    with open('users.csv', 'r', encoding='utf-8') as file:
        reader = csv.reader(file)
        for row in reader:
            if row[0] == username and row[1] == password:
                session['username'] = username
                session['password'] = password
                session['name'] = row[2]
                return redirect(url_for('dashboard'))

    return render_template('login.html', login_message='ユーザー名またはパスワードが正しくありません')

```

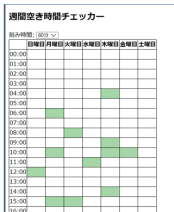
sessionに入れる

users.csv



username	password	name
1111	1111	あつし
2222	2222	きよし
3333	3333	あるばか
4444	4444	いのち
5555	5555	りんご

図 1: グラフ



各曜日の合計時間:

日曜日: 2.00 時間

月曜日: 3.00 時間

火曜日: 2.00 時間

水曜日: 2.00 時間

木曜日: 4.00 時間

金曜日: 2.00 時間

土曜日: 0.00 時間

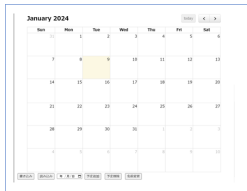
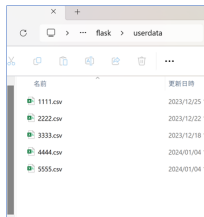
合計時間を保存

study_time.csv


username	Sun	Mon	Tue	Wed	Thu	Fri	Sat
2222	0	3	1	2	2	1	0
3333	1	1	1	1	3	0	0
1111	0	0	0	0	0	0	0
5555	0.25	1.5	0.5	1.75	1	0.75	0



図 2: グラフ

名前	更新日時
1111.csv	2023/12/25 1
2222.csv	2023/12/22 1
3333.csv	2023/12/18 1
4444.csv	2024/01/04 1
5555.csv	2024/01/04 1



4444.csv

ファイル 編集 表示

```

ID, Title, Start, End
event2, a, 2024-01-02, Invalid date
event1, b, 2024-01-18, Invalid date
event3, c, 2024-01-31, Invalid date
  
```

Pythonで作ったutf-8はエクセルで日本語が文字化けするのでメモ帳で

userdata/{username}.csv



図 3: グラフ

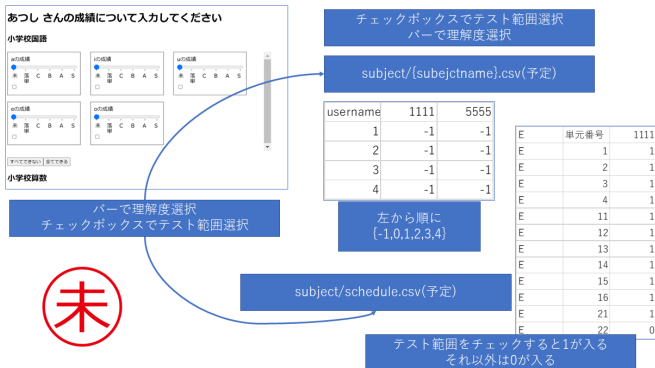


図 4: グラフ

2024年1月

前月

次月

1月

2020年

日	月	火	水	木	金	土
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

日付を選んでね！

2024-01

2024-02

2024-03

2024-01

2024-02

2024-03

2024-01

2024-02

2024-03

2024-01

2024-02

2024-03

2024-01

2024-02

2024-03

userdata/{username}/{subjectname}.csv(予定)

4444.csv

shakai.csv

X

+

ファイル

編集

表示

Day, Date, Day of Week, Unit, Study Hours, Description
1, 2024-01-09, Tuesday, 0, 0, 0, Start
1, 2024-01-09, Tuesday, 1, 3, 0, わたしたちの大好きなまち
2, 2024-01-10, Wednesday, 1, 1, 144943820247197, わたしたちの大好きなまち
2, 2024-01-10, Wednesday, 2, 2, 8550561797752803, はたらく人とわたしたちのくらし
3, 2024-01-11, Thursday, 2, 2, 021348314606743, はたらく人とわたしたちのくらし
3, 2024-01-11, Thursday, 4, 2, 978651685383257, わたしたちの市の歩み
4, 2024-01-12, Friday, 4, 0, 9224719101123613, わたしたちの市の歩み
4, 2024-01-12, Friday, 3, 4, 14494382022472, 地いきの安全を守る
4, 2024-01-12, Friday, 11, 1, 7067415730337063, 県の地図を広げて
4, 2024-01-12, Friday, 12, 0, 22584269662921064, 健康なくらしとまちづくり
5, 2024-01-13, Saturday, 12, 6, 844943820224723, 健康なくらしとまちづくり
5, 2024-01-13, Saturday, 16, 1, 1550561797752774, わたしたちの県のまちづくり
6, 2024-01-14, Sunday, 16, 3, 477520806876444, わたしたちの県のまちづくり
6, 2024-01-14, Sunday, 14, 2, 4382022471910116, 地域で受け継がれてきたもの
6, 2024-01-14, Sunday, 21, 4, 0842696629213435, 日本の国土とわたしたちのくらし
7, 2024-01-15, Monday, 21, 0, 7921348314606798, 日本の国土とわたしたちのくらし
7, 2024-01-15, Monday, 15, 1, 2078651685383202, 昔から今へと続くまちづくり
8, 2024-01-16, Tuesday, 15, 1, 9617977528089945, 昔から今へと続くまちづくり
8, 2024-01-16, Tuesday, 13, 1, 0382022471910055, 自然災害にそなえるまちづくり
9, 2024-01-17, Wednesday, 13, 1, 4000000000000061, 自然災害にそなえるまちづくり
9, 2024-01-17, Wednesday, 999, 0, 0, End

仮のhtml
一科目分しか反映できていない



図 5: グラフ

スケジュールの作り方

subject/schedule.csvで必要な
単元のみ抜き取る

E	単元番号	1111
E	1	0
E	2	0
E	3	0
E	4	0
E	11	1
E	12	1
E	13	1
E	14	1
E	15	1

C:\Users\nasut\Downloads\スケジュールv1\スケジュールv1>python sten.py
Target Unit Numbers: {'1', '11', '12', '21', '4', '15', '16', '3', '2', '13', '14'}

C:\Users\nasut\Downloads\スケジュールv1\スケジュールv1>python sten.py
Target Unit Numbers: {'1', '11', '12', '14', '15', '16', '21', '23', '24', '4', '1', '2', '13'}

NAME	CODE	PREDECESSORS	SUCCESSORS	DAYS	ES	EF	LS	LF	SLACK	CRITICAL
Start	0		['1', '2', '13']	0	0	0	0	0	0	YES
わたしたちの大団円なま	1	['0']	['4']	17	0	17	0	17	0	YES
わたしたちの大団円なま	2	['0']	['13']	20	0	20	17	20	3	NO
わたしたちの大団円なま	3	['0']	['12']	17	0	17	20	17	0	YES
わたしたちの大団円なま	4	['1']	['10', '11']	16	17	33	17	33	0	YES
わたしたちの大団円なま	11	['2', '4']	['14']	7	33	40	33	40	0	YES
わたしたちの大団円なま	12	['3']	['12']	20	17	40	40	40	0	YES
わたしたちの大団円なま	13	['12']	['1000']	10	40	50	40	50	0	YES
わたしたちの大団円なま	14	['4']	['10']	10	33	43	33	43	0	YES
わたしたちの大団円なま	15	['14']	['1000']	13	43	56	43	56	0	YES
わたしたちの大団円なま	16	['11']	['12']	17	40	57	40	57	0	YES
わたしたちの大団円なま	21	['14']	['1000']	20	40	59	39	59	0	YES
End	999	['13', '15', '21']		0	59	59	59	59	0	YES

抜き取った単元範囲以外
の前単元、あと単元をす
べて削除

単元番号が0の"Start",
999の"End"を作成

前単元が存在しなくなっ
たものはすべて"Start"を
前単元にする

後単元が存在しなくなっ
たものはすべて"End"を
後単元にする

図 6: グラフ

LS列を抜き出してこの順番にスケジュールに格納していく

LS	DAYS
0	0
0	17
13	20
23	17
17	16
33	7
40	29
69	10
56	10
66	13
40	19
59	20
79	0

※なぜLSなのか？
LS = 最遅開始時間であり、この時間に遅れることで全体のスケジュールが遅延してしまう
= 優先度と考えられるのではないかな？
基本的に実行順になっており、floatなどと比べると並べやすい

DAYSを比率として考え、合計時間を分割し、単元ごとの合計時間を算出
指導時数の合計を理想勉強時間とする

username	Sun	Mon	Tue	Wed	Thu	Fri	Sat
2222	0	3	1	2	2	1	0
3333	1	1	1	1	3	0	0
1111	0	0	0	0	0	0	0
5555	0.25	1.5	0.5	1.75	1	0.75	0

4444.csv
ファイル 編集 表示
ID, Title, Start, End
event2, a, 2024-01-02, Invalid date
event1, b, 2024-01-18, Invalid date
event3, c, 2024-01-31, Invalid date

study_time.csv
userdata/[username].csv
を使用して計算し、
今日から最も近いイベントまでの期間までの合計勉強時間を計算

各単元ごとの合計時間:
0: 0.0 時間
1: 4.14494382022472 時間
2: 4.876604049382023 時間
4: 3.9013235955066183 時間
3: 4.14494382022472 時間
11: 1.79479115730337983 時間
12: 7.079786516853933 時間
16: 4.632584249662922 時間
14: 4.43262622071918116 時間
21: 4.876604049382023 時間
18: 3.1696429233483147 時間
13: 2.43262622071918116 時間
999: 0.0 時間
理想勉強時間: 178
合計勉強時間: 43.480000000000006 時間
圧縮率: 4.1813820204792625

圧縮率
= 理想勉強時間
合計勉強時間

圧縮率を出すことで、
どれだけ無理のある
スケジュールなのか
を提示する

図 7: グラフ


```
for unit in unit_to_study:
    if unit not in unit_study_time:
        continue

    if study_hours > unit_study_time[unit]:
        study_schedule.append({
            "Day": day + 1,
            "Date": current_date.strftime("%Y-%m-%d"),
            "Day of Week": day_of_week,
            "Unit": unit,
            "Study Hours": unit_study_time[unit],
            "Description": unit_ratios_off[unit_ratios_off["CODE"] == unit][1]["DESCR"].values[0],
        })

    study_hours -= unit_study_time[unit]
    del unit_study_time[unit]
else:
    study_schedule.append({
        "Day": day + 1,
        "Date": current_date.strftime("%Y-%m-%d"),
        "Day of Week": day_of_week,
        "Unit": unit,
        "Study Hours": study_hours,
        "Description": unit_ratios_off[unit_ratios_off["CODE"] == unit][1]["DESCR"].values[0],
    })

unit_study_time[unit] -= study_hours
break
```

日に単元を追加していく

if(日の勉強時間 > 単元の勉強時間)
日の単元に追加
単元を削除
次の単元に進む
else(単元の勉強時間 > 日の勉強時間)
日の単元に追加
その日は終了

4444.csv shakai.csv shakai.csv

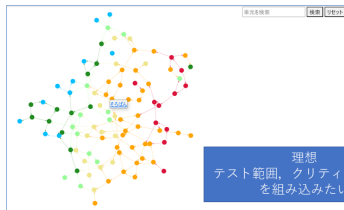
ファイル 編集 表示

```
Day,Date,Day of Week,Unit,Study Hours,Description
1,2024-01-09,Tuesday,0,0,0,Start
2,2024-01-09,Tuesday,1,3,0,わたしたちの大好きなまち
3,2024-01-10,Wednesday,1,1,144943820247197,わたしたちの大好きなまち
4,2024-01-10,Wednesday,2,2,850561797752803,はたらく人とわたしたちのくらし
5,2024-01-11,Thursday,2,2,801346514666143,はたらく人とわたしたちのくらし
6,2024-01-11,Thursday,4,2,97865168538257,わたしたちの昔の歩み
7,2024-01-12,Friday,4,0,927419101123615,わたしたちの昔の歩み
8,2024-01-12,Friday,3,4,1449438202472,地いきの安全を守る
9,2024-01-12,Friday,11,1,7067415730317083,雨の雨国を上げて
10,2024-01-12,Friday,12,0,25584360462921084,健康なくらしとまちづくり
11,2024-01-13,Saturday,12,6,84494382024723,健康なくらしとまちづくり
12,2024-01-13,Saturday,10,1,150561797752714,わたしたちの昔のまちづくり
13,2024-01-14,Sunday,16,3,4775290698976444,わたしたちの昔のまちづくり
14,2024-01-14,Sunday,14,2,4318202471910116,地域で受け継がれてきたもの
15,2024-01-14,Sunday,21,4,0842696629213435,日本の国とわたしたちのくらし
16,2024-01-15,Monday,21,0,7921348314666798,日本の国とわたしたちのくらし
17,2024-01-15,Monday,16,1,207865168538202,昔から今へと続くまちづくり
18,2024-01-16,Tuesday,13,1,861797752802846,昔から今へと続くまちづくり
19,2024-01-16,Tuesday,13,1,038202471910055,自然災害にそなえるまちづくり
20,2024-01-17,Wednesday,13,1,400300000000061,自然災害にそなえるまちづくり
21,2024-01-17,Wednesday,999,0,0,End
```

図 8: グラフ



subject/{username}/{subjectname}.html(予定)



理想
テスト範囲、クリティカルパス
を組み込みたい

Flask内でルートを作り、リアルタイムでの
グラフ作成が可能なことは確認した



図 9: グラフ

清水さんの教材作成画面
キーワード抽出

手を付けられてない
Flaskのルートの一般化がよくわからない



図 10: グラフ