



NATIONAL UNIVERSITY SINGAPORE

REASONING SYSTEM PROJECT

YOSS

Yoga Class Scheduling System

Group Members

LI DUO

LIM CHONG SENG HERMANN

LU JIAHAO

YAM GUI PENG DAVID

ZHAO YAZHI

Student ID

A0195364W

A0195392U

A0091835Y

A0195315A

A0195305E

Supervised by

GU Zhan (Sam)

NUS-ISS

Sunday 21st April, 2019

Executive Summary

With the overwhelming welcome for the yoga activity, A * yoga has opened its 5th new yoga studio in Singapore recently. It targets to provide 100+ different styles of yoga classes every week with experienced yoga masters and well decorated yoga rooms to customers. However, with the various yoga classes, complicated yoga master skills and available dates, it is very troublesome to plan a weekly yoga class schedule.

Our group, containing 5 part-time students under reasoning system course, decided to take this chance to help A * yoga studio by applying knowledge we have learnt from the course. Hence, our group developed an automated scheduling system, Yoga Class Scheduling System (YOSS), to generate a weekly yoga class schedule based on the default system data or data provided by user.

Inside YOSS, we first set out to perform knowledge acquisition by interviewing a subject matter expert. To build the system, we decided to utilize Search and Rule-Based Reasoning as tools through Django web framework and KIE server to perform the scheduling task. What's more, we have come out with a calendar user interface to display the scheduling results to users.

Our team learned a lot in the process of working on this project. We got the chance to apply techniques, such as search algorithm and rule-based reasoning, which we learned in our lectures and workshops in this viable business application scenario, and also picked up technical skills which would surely prove useful in the future course of our work.

Contents

1. Business Problem Background	4
2. Objectives & Success Measurements	4
a. Objectives	4
b. Success Measurements	4
I. Knowledge Model	4
II. Knowledge Identification	5
III. Knowledge Acquisition	5
IV. Knowledge Specification	7
V. Knowledge Refinement	8
3. Solution	9
a. System Architecture	9
b. Assumptions	9
I. Eligibility of Users	9
II. Knowledge Acquisition	9
c. System Features	10
I. Highly Personalized	10
II. Ease of Access	10
d. Limitations	10
4. Performance & Validation	10
5. Conclusions	14
6. Future Work	15
7. References	15
Appendices	15

1. Business Problem Background

Many physical activities that are good for your health also create serious imbalances in the body. However, yoga is one of the few physical activities that promotes whole body wellness – both physically and psychologically – without creating the imbalances that lead to injuries and complications later on in life.

A * yoga has a wealth of yoga studios across Asia, with prime locations in Singapore. A * Yoga studios offer over 100 classes each week in a diversity of yoga practices taught by a community of over 15 passionate and dedicated yoga masters.

A * yoga has opened its 5th new yoga studio in Singapore recently. A * yoga targets to provide around 112 different styles of yoga classes every week, 15 experienced yoga masters and 4 well-decorated yoga rooms to customers. However, with the various yoga classes, complicated yoga master skills and available dates, it is very troublesome for a person to map and come out with the daily or weekly yoga class schedule.

2. Objectives & Success Measurements

a. Objectives

The objective of this project is to create an automated **Yoga Class Scheduling System (YOSS)** that will generate a daily or weekly yoga class schedule to the system users based on the default data or data provided by user. YOSS makes use of **Search** and **Rule-Based Reasoning**, to assign yoga masters based on their available time slots and skill sets, yoga class, and yoga room to time slots. As YOSS allows to upload new data, it should be applicable to almost all studios for A * yoga.

The primary target audience of our system are Asia Square Branch, A * yoga studio. However, our system can also very much be utilized by other branches or even, other studios as a tool for planning and scheduling.

By using the default dataset provided by the system, YOSS will be able to analyse and generate a weekly yoga class schedule with assigned yoga master, room and timeslots. In other way, system users can upload latest data into the system to get updated schedule.

b. Success Measurements

I. Knowledge Model

1. Knowledge Identification
2. Knowledge Specification
3. Knowledge Refinement

Different tasks and components are done at each stage to derive the final output of yoga class schedule.

II. Knowledge Identification

Knowledge identification sets the groundwork for the next stage encompassing knowledge specification. In the context of building a scheduling system for yoga class, two main sources have been identified and are documented in Table 1

S/N	Source of Information	Insights from Information Sources	Knowledge acquisition technique
1	A * yoga studio	It provides basic information on yoga rooms, yoga timeslots, yoga masters and yoga classes.	Data gathering from available/documented information.
2	A * yoga administration Expert	The subject matter expert, will be able to: 1. Identify and explain the considerations and constraints when plan class schedule. 2. Validate or further reinforce our understanding from the available data.	Elicitation of tacit knowledge through the conduct of interview

Table 1: knowledge Source and Acquisition Technique

III. Knowledge Acquisition

Following from the identification of knowledge sources, knowledge acquisition is conducted to capture the problem-solving domain knowledge. The knowledge has been describe in Table 1 and the corresponding results are presented using a domain diagram as shown in Figure 1.

Yoga Class Scheduling System Domain Model

leleljh | April 21, 2019

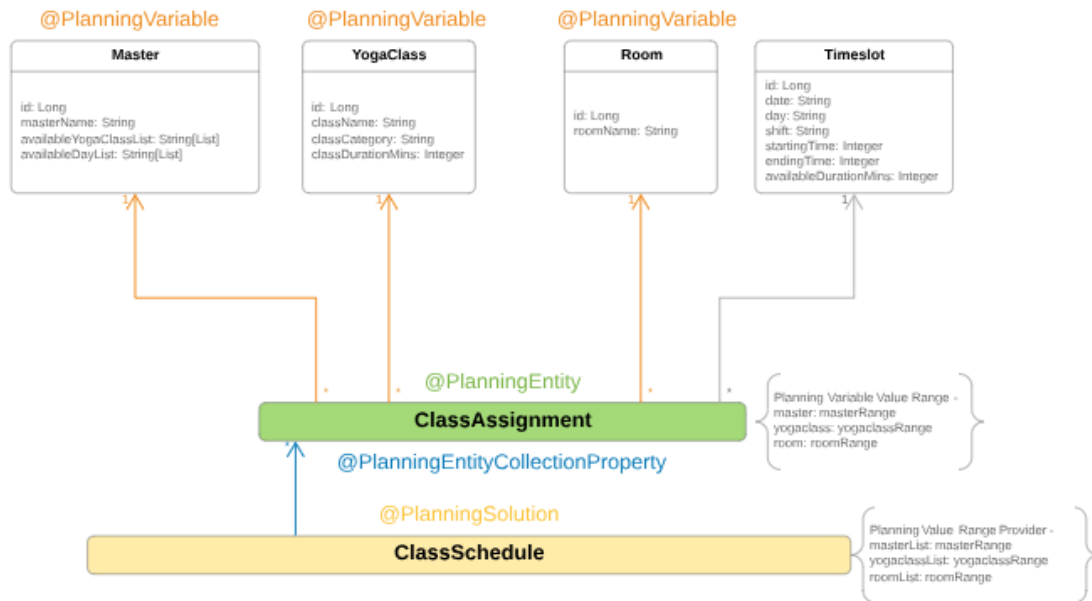


Figure 1: Domain Model

The domain model shows all objects, relationship(s) and direction(s) clearly. It contains one planning entity Class Assignment, one planning solution Class Schedule, three planning variables (Master, Yoga Class, and Room), as well as the relationships among them. All planning variables have a one-to-many relationship with the planning entity Class Assignment.

Table 2 illustrates an example using the Domain Diagram in Figure 1.

S/N	Category	Information
1	Planning Variable(s)	A yoga Master object with his/her name, available dates and available skill sets. A YogaClass object with it's name, category and duration. A yoga Room object with it's name.
2	Planning Entity	Class Assignment connects the Planning Variables to the Problem Property. Hence, it includes the planning variables: yoga Master , YogaClass , and yoga Room . It also includes the Problem Property Timeslot .
3	Planning Solution	Class Schedule contains the full set of Planning Entities. (i.e. A weekly schedule with the yoga Master(s) , YogaClass(es) and yoga Room(s) assigned to all Timeslot(s)).

Table 2: An example of Domain Diagram

IV. Knowledge Specification

There are a total of 12 rules (8 hard & 4 soft).

The hard rules were chosen either because they conflict with reality (e.g. one resource assigned to two conflicting Timeslot(s), classes assigned to Timeslot(s) with not enough time) or because they are not considered valid from the business perspective (e.g. no “Hot” yoga classes on weekday nights).

The Soft rules were chosen to further direct the solver to a more “optimal” solution (e.g. reducing the unique shifts of Masters)

The rules and their descriptions are as follows:

S/N	Category	Type	Description
1	RuleConflictMasterTimeslot	Hard	Master cannot be assigned to Timeslot(s) with overlapping timeslot durations
2	RuleConflictRoomTimeslot	Hard	Room cannot be assigned to Timeslot(s) with overlapping timeslot durations
3	RuleConflictClassTimeslot	Hard	Available duration of the Timeslot must be more than the YogaClass duration
4	RuleMasterUnavailable	Hard	Master cannot be assigned to a day where he/she is unavailable
5	RuleMasterUnskilled	Hard	Master cannot be assigned to a class that he/she is unable to teach
6	RuleUnassignedYogaClass	Hard	All Timeslot(s) must have an assigned Master, YogaClass & Room

7	RuleNightNoHotYoga	Hard	Every weekday, YogaClass belonging to the "Hot" category does not appear at least once at night
8	RuleSameCategoryBackToBackRepeat	Hard	YogaClass with the same category cannot repeated at back to back Timeslot(s)
9	RuleSameClassRepeatInDay	Soft	YogaClass with the same name should not repeat more than 1 time in a day
10	RuleSameDayMasterMultipleShift	Soft	Master should not have more than 1 shift in the same day
11	RuleSameDayMasterLargeMinMaxSlotIndex	Soft	Master should not be assigned to YogaClass(es) that are too far apart in the day (e.g. Only assigned 1 YogaClass at the start of the day and 1 YogaClass at the end of the day)
12	RuleClassCategoryNotSpreadOut	Soft	YogaClass with the same Category should be spread out over the week (i.e. should not all be in 1-2 days)

V. Knowledge Refinement

The knowledge refinement would be an iterative process and it includes model validation and model refinement. For model validation, different sets of test data will be used to run simulation, and the result will be compared with logic constraints.

For model refinement, certain assumptions, adjustments and pre-requirement is included into the model after validation of the model results and discussion with staff from A * yoga studio.

3. Solution

a. System Architecture

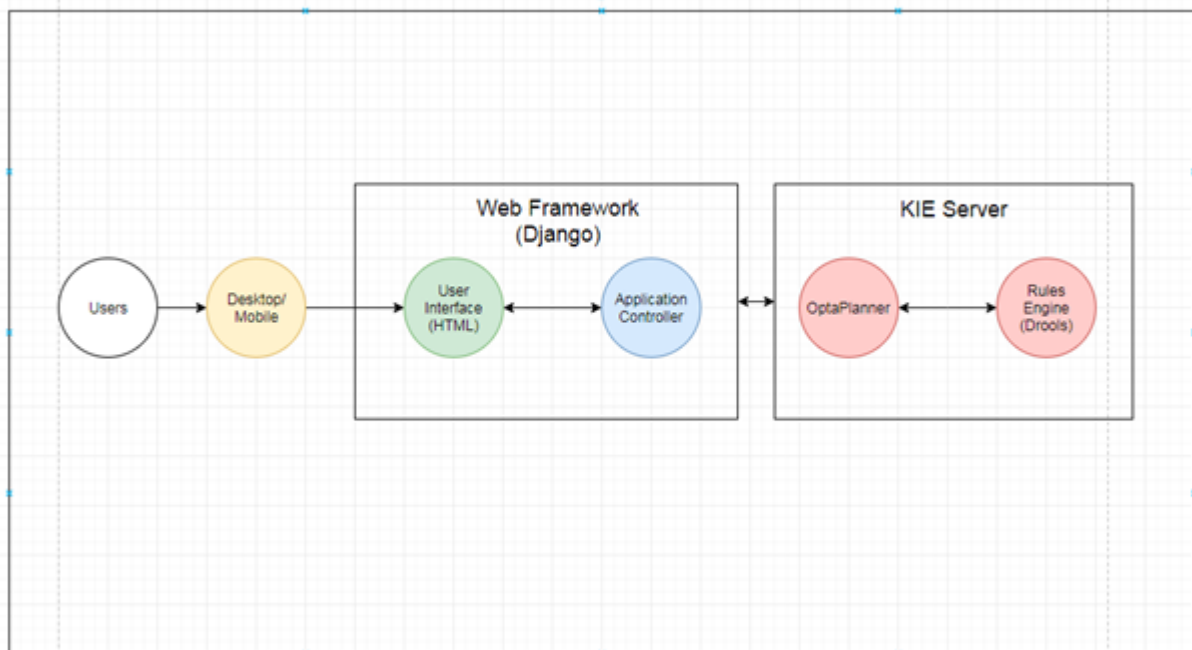


Figure 2: System Architecture Diagram

Figure 2 shows the system architecture diagram of YOSS. It illustrates how the different components interact with each other through the Django web framework and KIE server. After system user uploads excel formatted data into the User Interface, it will be passed through the Django framework to KIE server via API call. Within the KIE server framework, drools, as the rules engine, will connect OptaPlanner library to perform the reasoning portion of the system, for example the hard and soft constraints of the class scheduling. The rules engine will then return the results, with the final schedule being passed back to the HTML page to be displayed.

b. Assumptions

I. Eligibility of Users

The major target audience would be staffs working in A * yoga studios. User need to provide and upload prepared excel data through user interface. User will be able to come out a weekly yoga class scheduling plan as the result of the system.

II. Knowledge Acquisition

Data for yoga masters, yoga rooms, yoga class and yoga time slots are from A * yoga studio, Asia Square branch. It is assumed the data provided by the studio is correct.

c. System Features

I. Highly Personalized

YOSS is highly personalized for each studio with provided data.

Each studio have its own yoga masters, yoga rooms and yoga classes. YOSS is able to come out with a personalized schedule for each studio with its own dataset.

II. Ease of Access

YOSS is built on a web-based application. After the deployment on a server, the user can easily access it from a browser with Internet connection.

Based on the bootstrap framework, YOSS is mobile friendly and can dynamically fit different screen sizes. Users can use it directly from their mobile phones. If the studio wants to use YOSS result as a service in their branches, only a tablet is needed for on-site customers.

d. Limitations

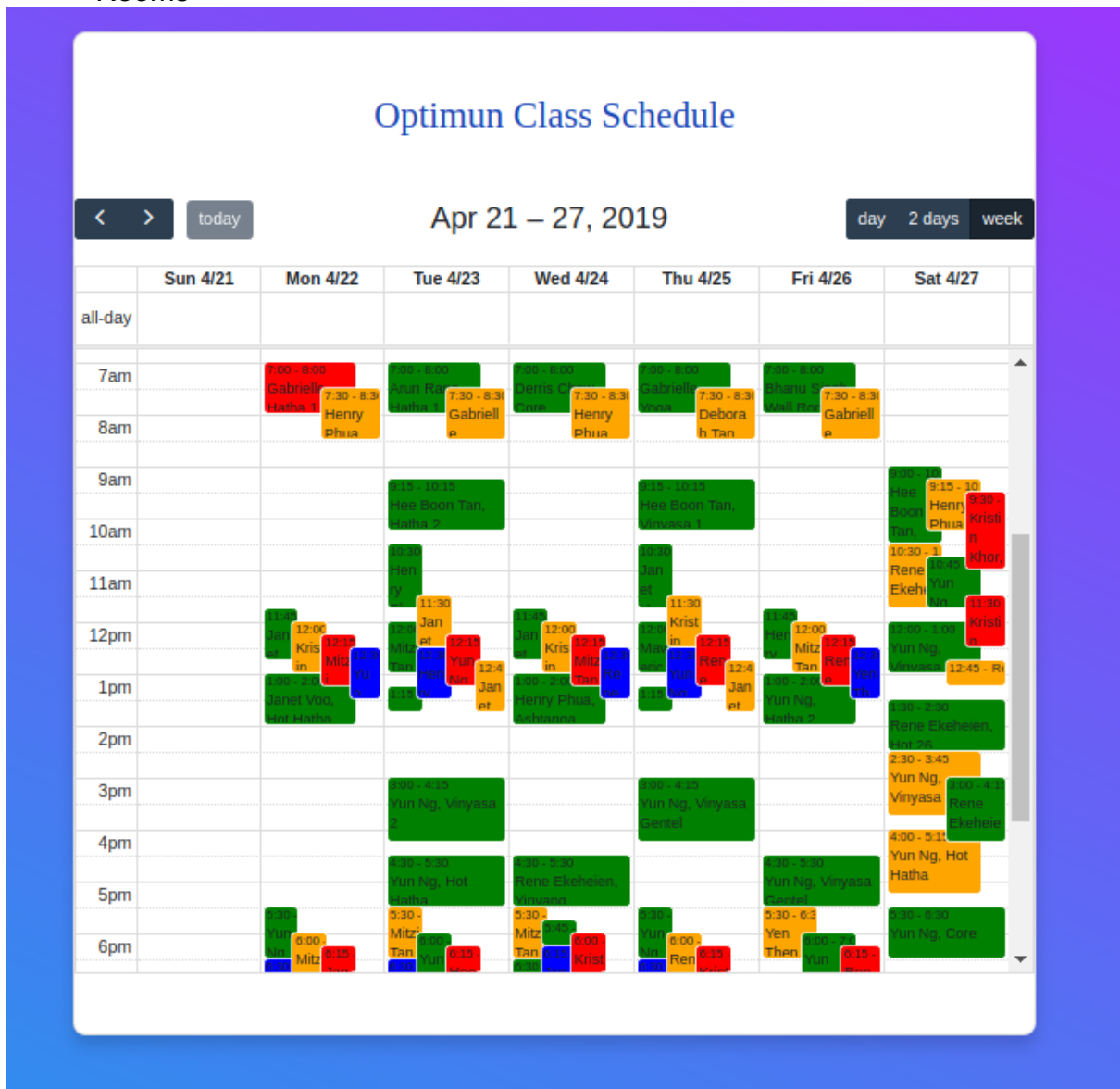
As this project is mainly target to solve the scheduling problem for A * yoga studio, Asia Square branch, our group gets the specific rules directly from the staff that working in the yoga studio branch. Therefore, constraints and weight of each constraint are predefined and fixed in the KIE server in YOSS.

To simplify the data preparation process, we request user to upload data based on required data format. The system may reject user request if the data is not in required formatting.

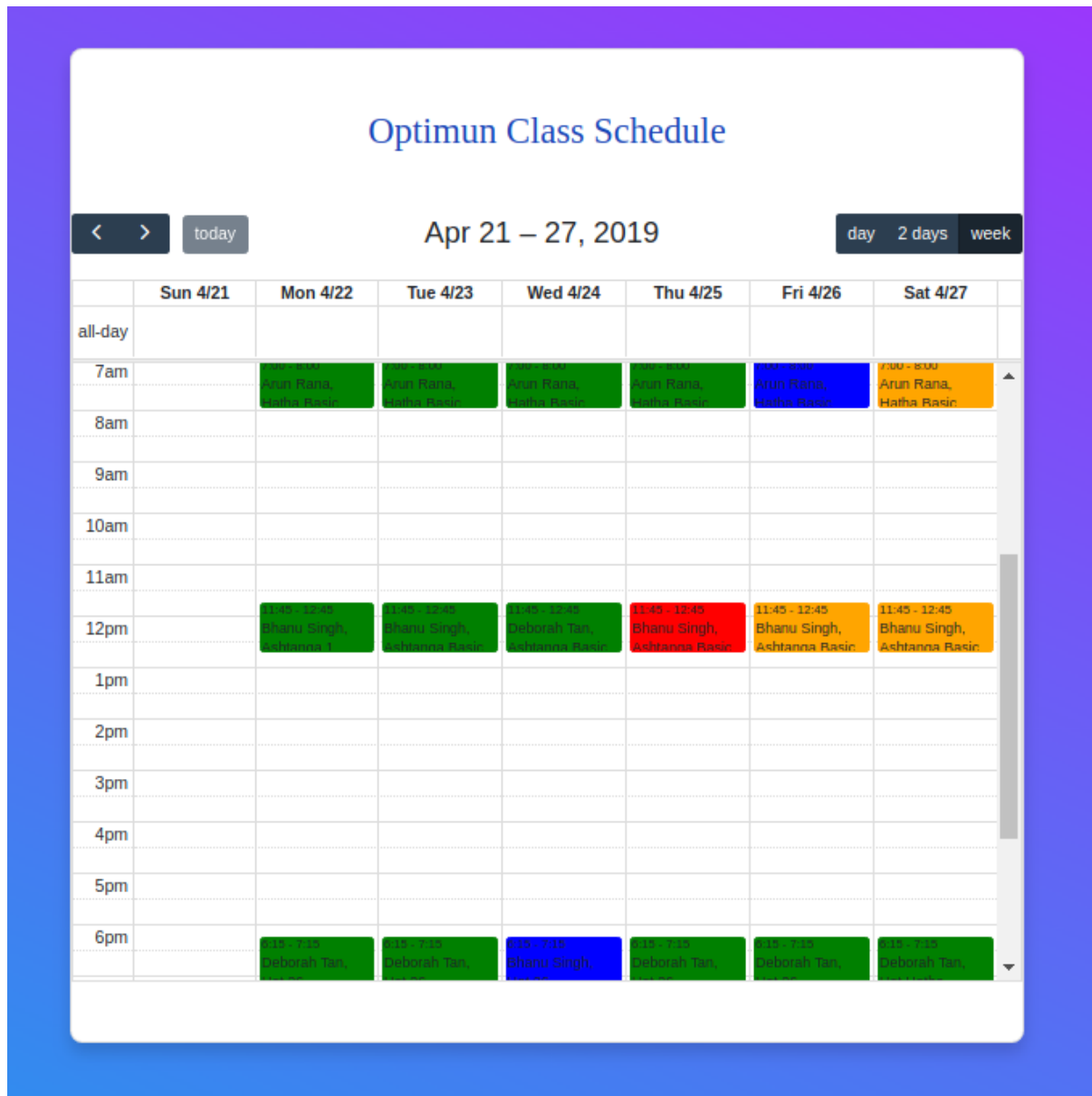
4. Performance & Validation

We have performed validation on 3 different scenarios to ensure that YOSS provides the correct expected output. Scenario 1 is the full set of data from A * yoga studio, it takes roughly 3-5mins to run. Scenario 2 is to test for generating a weekly yoga class schedule with time slots spaced out and all yoga masters available. Scenarios 3 is on the extreme case where user does not follow the instruction to upload data in correct formatting.

Scenario 1 - (Full dataset) 112 Timeslots, 29 Yoga Classes, 14 Masters, 4 Rooms



Scenario 2 - 21 Timeslots (3 each day at fixed timings), 8 Yoga Classes, 3 Masters (available for all timeslots and classes), 4 Rooms



Scenario 3 - Wrong input format:

1. Changing column names (e.g. master name instead of masterName)
2. Changing day names from 3 letters (e.g. Thu renamed as Thursday/ Thur)
3. Wrong input availableClassList/ availableDayList, requires “ , “ between every class name
(e.g. availableClassList written as Hatha Basic,Hatha 1 Hatha 2 instead of Hatha Basic, Hatha 1, Hatha 2)

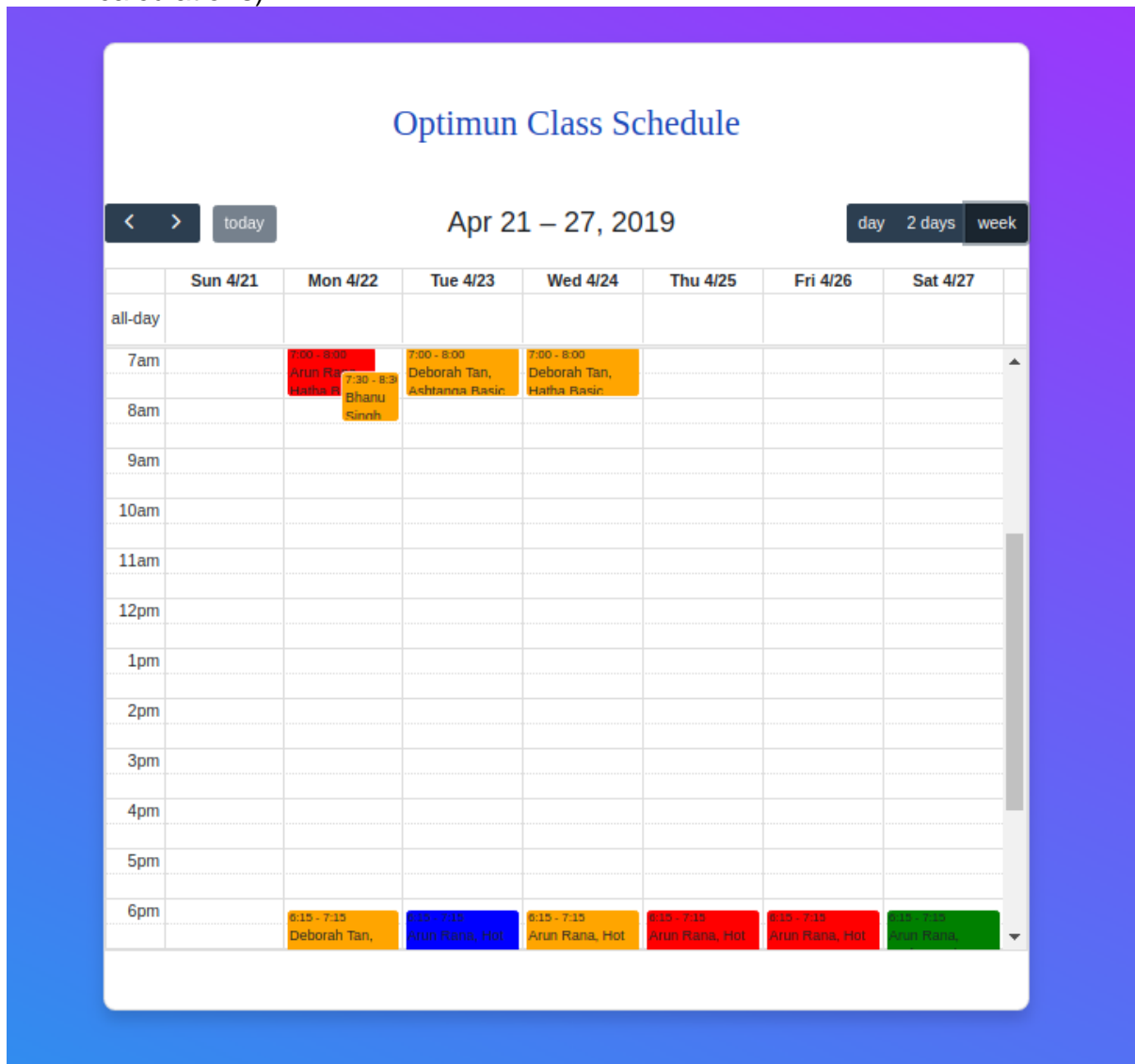
3.1 If column names are changed:

```
← → ↻ 127.0.0.1:8000/calendar/
GitHub Kaggle Scikit Learn Keras R RStudio KD Nuggets fast.ai Translate PDF PDF Binder Google KIE WB

KeyError at /calendar/
'masterName'

Request Method: POST
Request URL: http://127.0.0.1:8000/calendar/
Django Version: 2.1.5
Exception Type: KeyError
Exception Value: 'masterName'
Exception Location: pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item, line 1218
Python Executable: /home/iss-user/anaconda3/bin/python
Python Version: 3.6.8
Python Path: ['/home/iss-user/MTech19/IRS-RS-2019-03-09-151PT-GRP-Y055/SystemCode',
              '/opt/ros/kinetic/lib/python2.7/dist-packages',
              '/home/iss-user/anaconda3/lib/python3.6.zip',
              '/home/iss-user/anaconda3/lib/python3.6',
              '/home/iss-user/anaconda3/lib/python3.6/lib-dynload',
              '/home/iss-user/anaconda3/lib/python3.6/site-packages']
Server time: Sun, 21 Apr 2019 14:33:05 +0000
```

3.2 If day names are changed
(There will still be output, but “Monday” is not recognised properly in the rules calculations):



3.3 If wrong input availableClassList/ availableDayList
(class/ day will not be recognised properly in the rule calculations):
[Same as 3.2]

5. Conclusions

Our project, YOSS, is an automated system that will generate a daily or weekly yoga class schedule based on the default system data or data provided by user. YOSS assigns yoga masters based on their available time slots and skill sets, yoga class, and yoga room to weekly time slots. By using YOSS, staff from A * yoga studio, Asia Square branch can easily come out with the weekly schedule. It helps on saving time and reducing human errors.

6.Future Work

However, with the limited time, our project scope is diminished to a minimum viable product (MVP) level, which is version 1.0. Here is a list of tasks for commercial version:

- To allow user to create, modify and delete constraints
- To allow user to modify weight of each constraint
- To design a user interface to allow user to key in input data instead of uploading files

7.References

[1] <http://www.optaplanner.org/learn/slides/optaplanner-presentation/training.html#/5/21>

[2] <https://fullcalendar.io/>

Appendices

User Manual

1 System Overview

Yoga Class Scheduling System (YOSS) makes use of **Search** and **Rule-Based Reasoning**, to assign yoga masters based on their available time slots and skill sets, yoga class, and yoga room to time slots. As YOSS allows to upload new data, it should be applicable to almost all studios for A * yoga.

The primary target audience of our system are Asia Square Branch, A * yoga studio. However, our system can also very much be utilized by other branches or even, other studios as a tool for planning and scheduling.

By using the default dataset provided by the system, YOSS will be able to analyze and generate a weekly yoga class schedule with assigned yoga master, room and timeslots. In other way, system users can upload latest data into the system to get updated schedule.

2 Supported browsers

2.1 Mobile devices

It supports the latest versions of each major platform's default browsers. Note that proxy browsers (such as Opera Mini, Opera Mobile's Turbo mode, UC Browser Mini, Amazon Silk) are not supported

	Chrome	Firefox	Safari	Android Browser & WebView	Microsoft Edge
Android	Supported	Supported	N/A	Android v5.0+ supported	Supported
iOS	Supported	Supported	Supported	N/A	Supported
Windows 10 Mobile	N/A	N/A	N/A	N/A	Supported

Table 1: Mobile Devices Support

2.2 Desktop browsers

The latest versions of most desktop browsers are supported.

	Chrome	Firefox	Internet Explorer	Microsoft Edge	Opera	Safari
Mac	Supported	Supported	N/A	N/A	Supported	Supported
Windows	Supported	Supported	Supported, IE10+	Supported	Supported	Not supported

Table 2: Desktop Browser Support

3 Deployment

Our system is deployed by Django, which is a Python-based free and open-source web framework. To run the system successfully, some basic packages are required with the latest version of Python 3 (i.e. Python 3.7.2).

Packages:

- Pandas
- Django
- Requests

Once the running environment is ready, please open the system folder in terminal, and run the command `python3 manage.py runserver`. You should see the following output on the command line:

```
Performing system checks...

System check identified no issues (0 silenced).

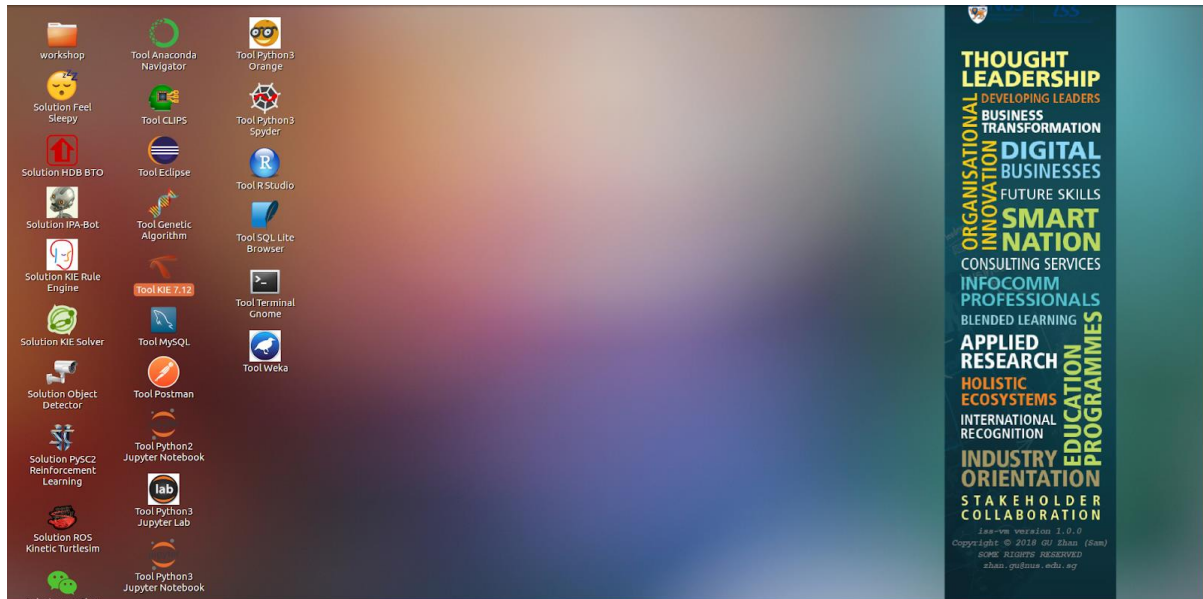
You have unapplied migrations; your app may not work properly
until they are applied.
Run 'python manage.py migrate' to apply them.

February 28, 2019 - 15:50:53
Django version 2.1, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Then you can go to the web browser and input URL "http://127.0.0.1:8000/" to use YOSS.

4 Procedure

Start KIE Workbench

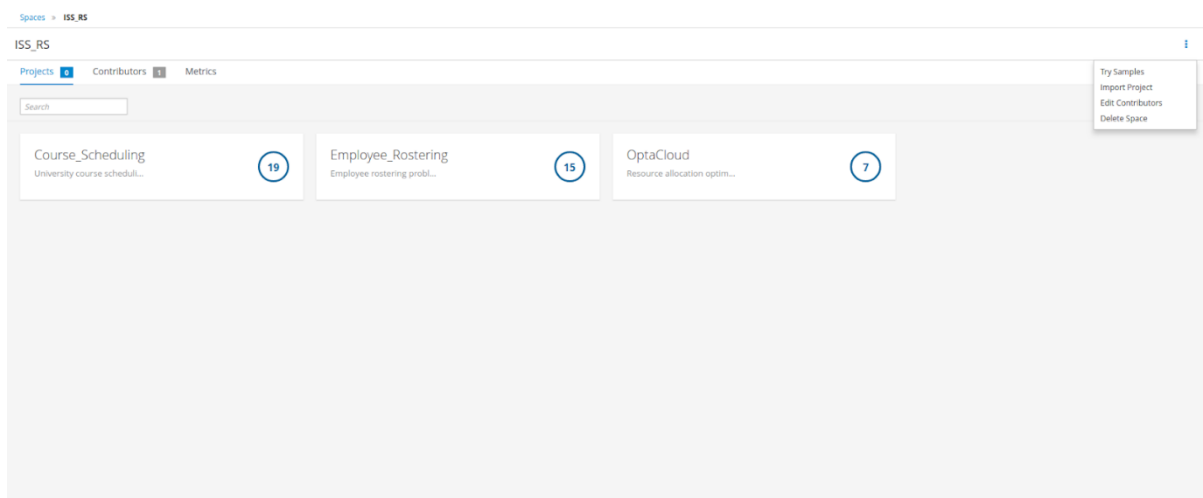


Sign into KIE Workbench as ID: wbadmin PW: wbadmin
(required as the django API request utilizes wbadmin/wbadmin for authentication)

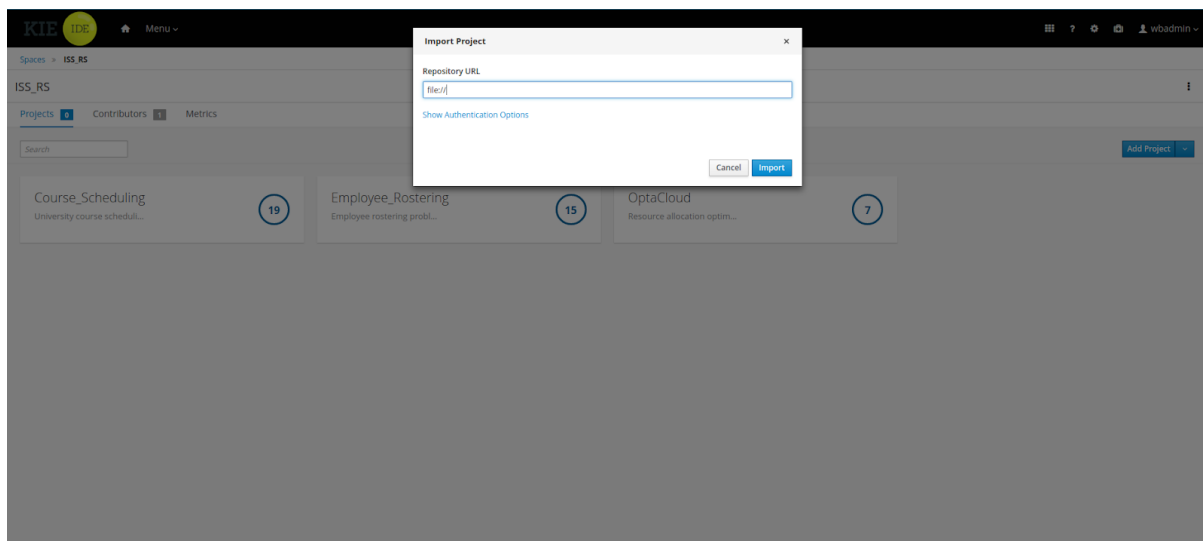


Unzip YogaClassScheduler1-2.

Open KIE Workbench, import YogaClasScheduler1-2



Ensure you type [file://<filepath to YogaClassScheduler1-2 folder>/YogaClassScheduler1-2](#)



For example:

Import Project

Repository URL

file:///media/sf_S-RS-Workshop-master/YogaClassScheduler-1-2

Show Authentication Options

CancelImport

Deploy the YogaClassScheduler-3 Project

Spaces > ISS_RS > YogaClassScheduler-3 > P: master

YogaClassScheduler-3

BuildDeployView Alerts

Assets16Contributors1MetricsSettings

All < Q

1-15 of 161 > of 2Import AssetAdd Asset

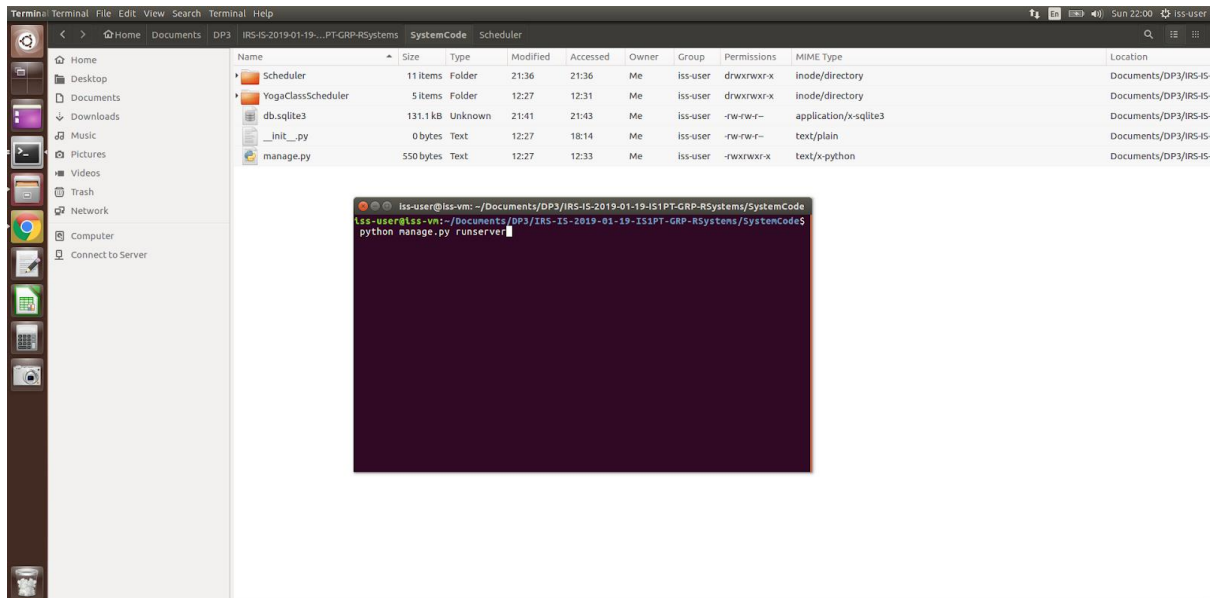
ClassAssignment	Data Objects	Last modified today	Created 1 week ago
ClassSchedule	Data Objects	Last modified today	Created 1 week ago
ClassScheduleScoreHolderGlobal	Globals Definitions	Last modified today	Created 1 week ago
Master	Data Objects	Last modified today	Created 1 week ago
Room	Data Objects	Last modified today	Created 1 week ago
RuleClassCategoryNotSpreadOut	Guided Rules	Last modified today	Created 1 week ago

Good job! You've finished starting up the KIE Server!

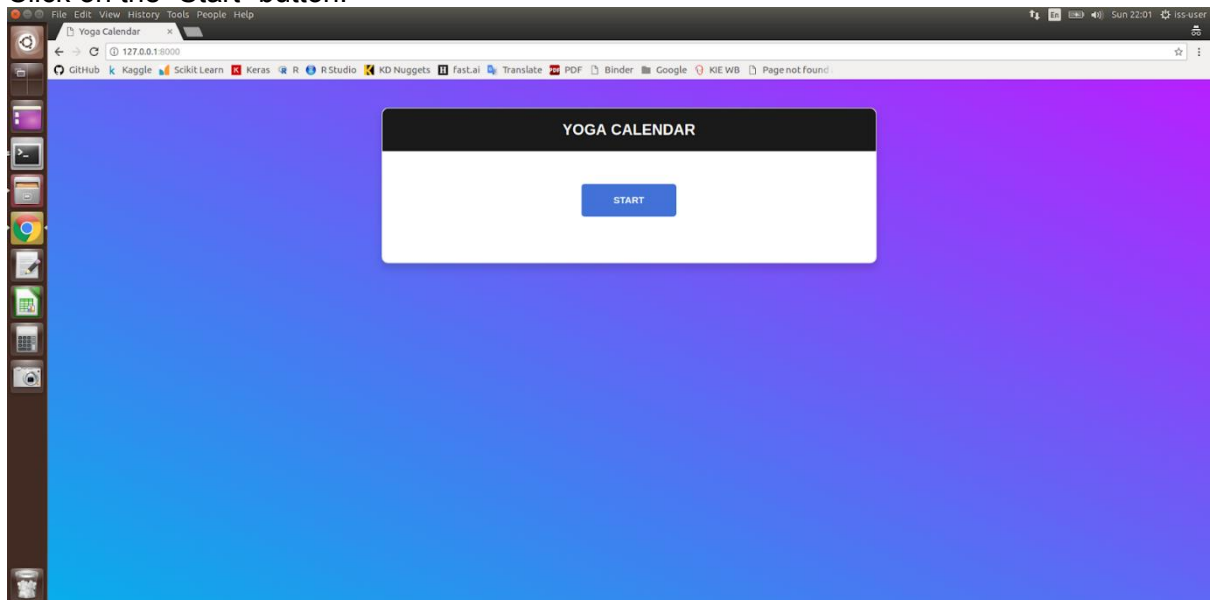
5 Using the system

Open up the Terminal console and set your current working directory to be “./IRS-RS-2019-03-09-IS1PT-GRP-YOS/SystemCode”.

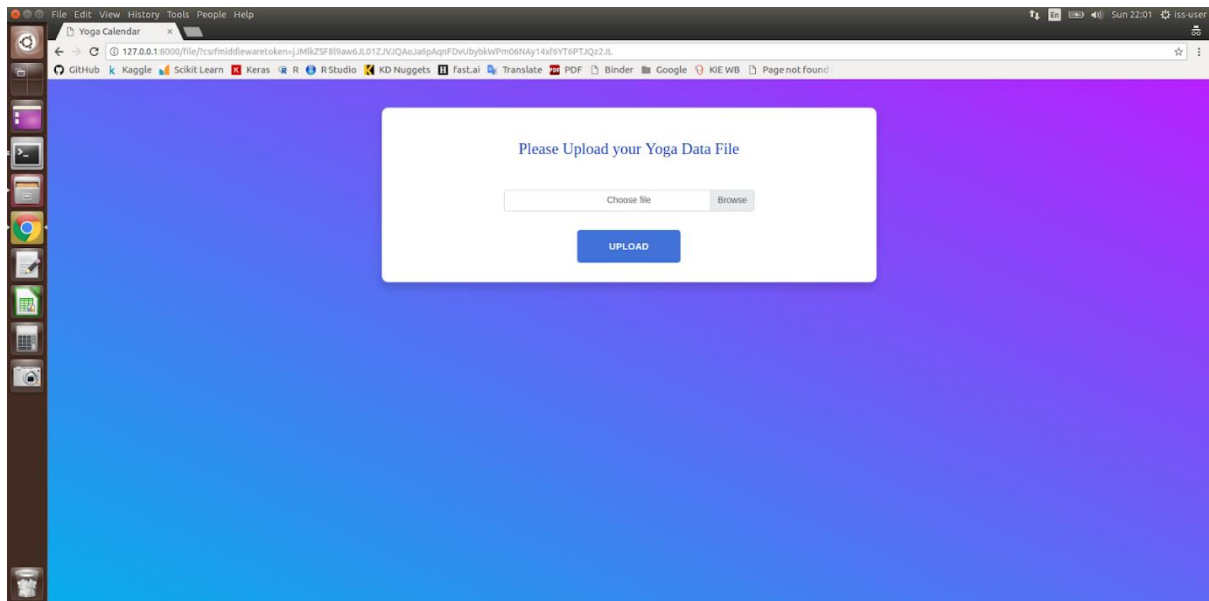
Type in the console: `python manage.py runserver`



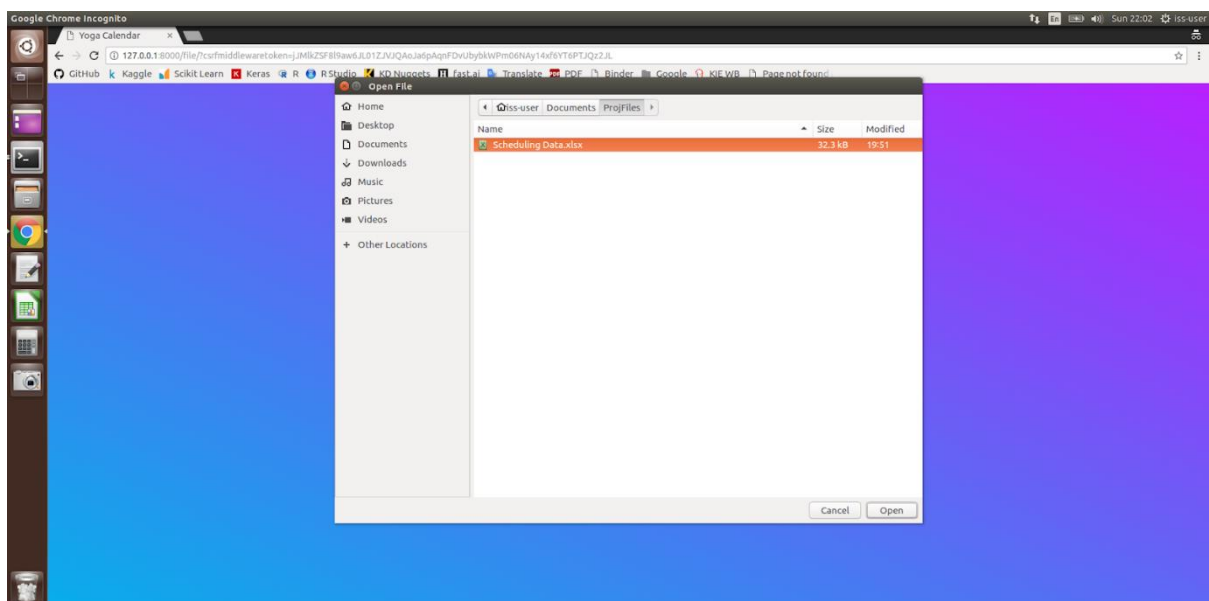
Open up your preferred browser and go to the URL “<http://127.0.0.1:8000>” as shown below. Click on the “Start” button.



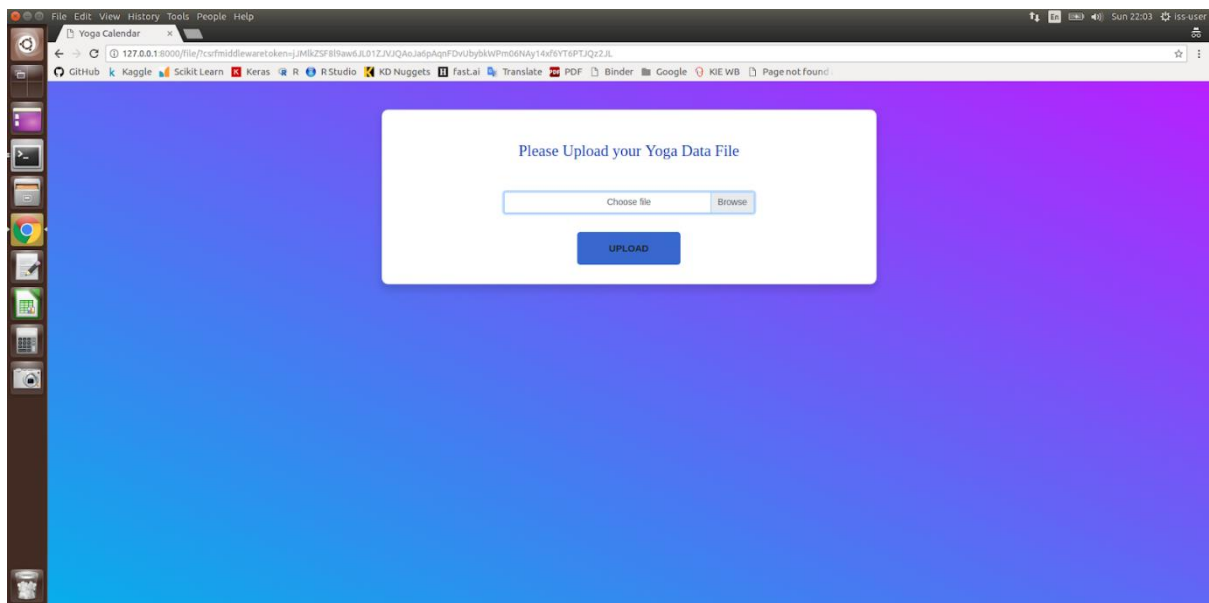
Click on the “Browse” button to select the Data Set excel file.



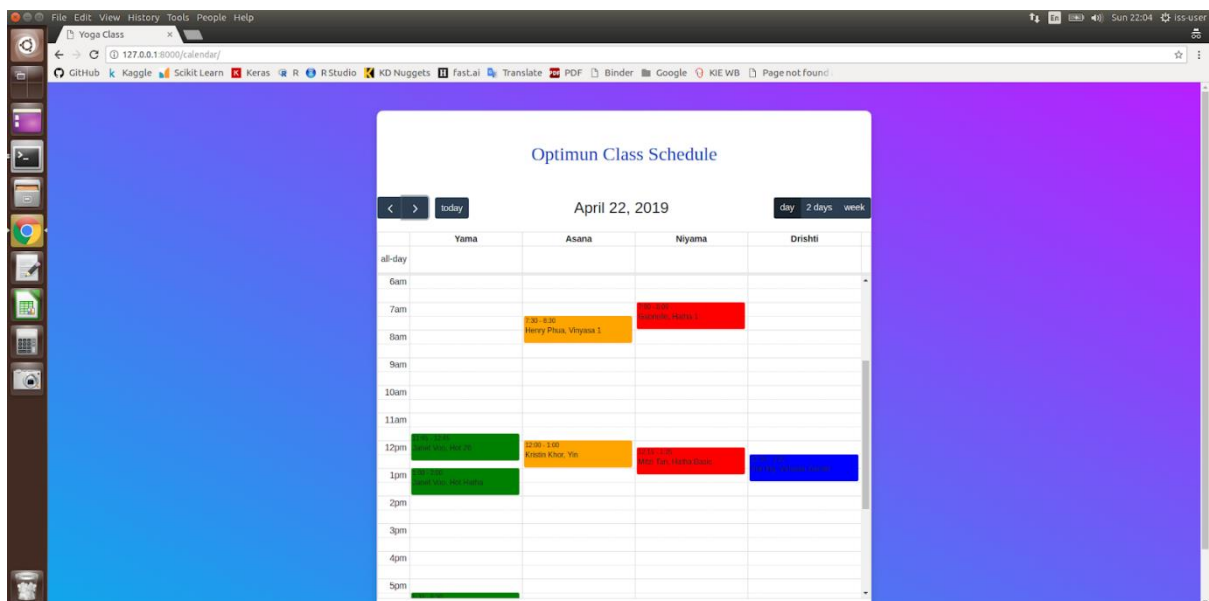
Select the Data Set excel file and click on the “Open” button.



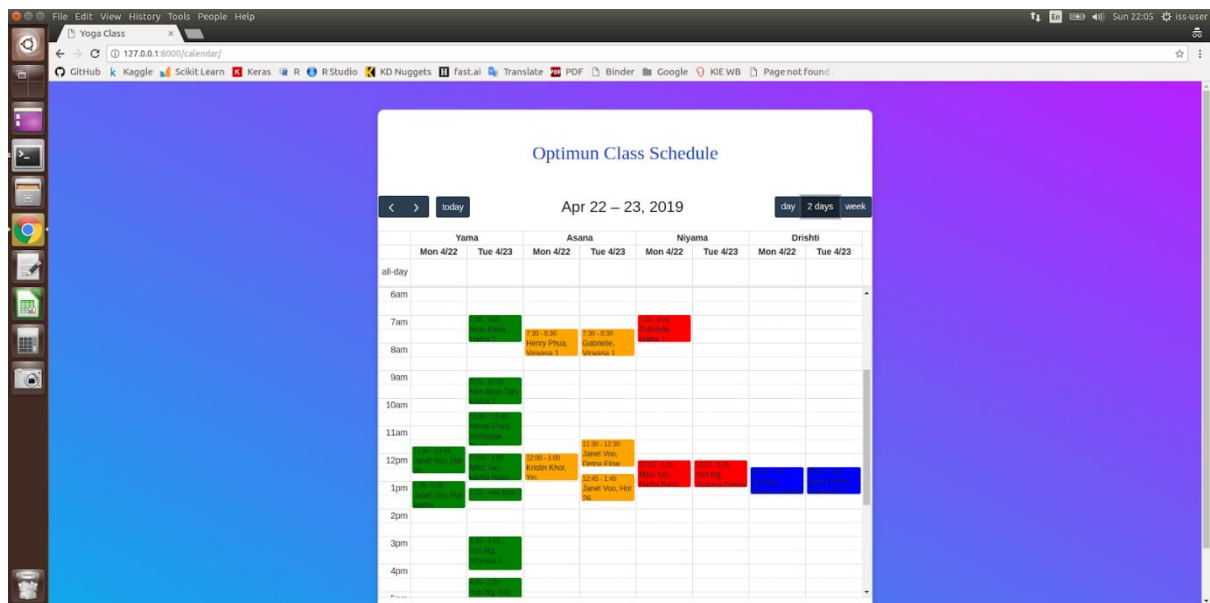
Click on the “Upload” button to start the solving process.



After a while, the calendar page will appear together with the solution. The default setting is the single day view. Views can be selected at the top right of the screen.



The view shown below is for the “2 days” view setting.



The view shown below is for the “week” view setting.

