Indian Institute of Information Technology, Dharwad

Automated Time Table Generator

User Manual

Developed by: Software Sages
Version: 1.0
Date: Nov, 2025
GitHub Repository link

Contents

1 Introduction

This project provides a complete automation suite for academic scheduling. It consists of three independent but interconnected Python modules:

- **timetable.py** — Generates balanced class timetables.

- **exam.py** — Generates exam schedules with room allocation and invigilation.

- **faculty.py** — Generates individual timetables for each faculty member.

All modules produce professional Excel outputs using the `openpyxl` engine and support multi-department, multi-semester scheduling.

2 System Requirements

2.1 Software Requirements

- Python 3.10+

- Required Python libraries:

  - `pandas`
  - `openpyxl`
  - `re` (built-in)
  - `datetime` (built-in)

2.2 Installation

Install dependencies:

```
pip install pandas openpyxl
```

3 Folder Structure

```
timetable.py
exam.py
faculty.py
data/
    rooms.csv
    faculty.csv
    time_slots.json
    courses*.csv
Balanced_Timetable_latest.xlsx
final_exam_schedule.xlsx
Faculty_Timetable_Grid_*.xlsx
```

4 Input File Specifications

4.1 rooms.csv

Mandatory columns:

- Room_ID

- Capacity

- Type

- Facilities

4.2 Course Files (courses*.csv)

Each course CSV must include:

- Course_Code

- Course_Title

- L-T-P-S-C

- Faculty

- Semester_Half

- Elective

- Students

4.3 time_slots.json

Example:

```
{
  "time_slots": [
    { "start": "09:00", "end": "10:00" },
    { "start": "10:00", "end": "11:00" }
  ]
}
```

5 Usage Instructions

5.1 Generate Class Timetable

Run:

```
python timetable.py
```

Output:

```
Balanced_Timetable_latest.xlsx
```

5.2   Generate Exam Schedule

Run:

```
python exam.py
```

Output:

```
final_exam_schedule.xlsx
```

5.3   Generate Faculty Timetables

Run:

```
python faculty.py
```

Output:

```
Faculty_Timetable_Grid_<timestamp>.xlsx
```

6   Module Descriptions

6.1   timetable.py — Class Timetable Generator

This module:

- Allocates course slots based on L/T/P hours.
- Ensures no faculty conflicts.
- Matches classrooms/labs based on capacity.
- Synchronizes elective baskets across groups.
- Avoids break time slots.
- Colors and merges identical adjacent cells in Excel.

6.1.1   Outputs

- Section-wise timetables
- Legends for each department
- Fully color-coded sheets

6.2   exam.py — Automated Exam Scheduler

Handles:

- Merged exam scheduling across departments.
- Max 1 exam/group/day.
- Max 4 total exams/day.

- Half-capacity rule for seating.

- Hall-last room allocation.

- Morning/afternoon elective splitting.

- Automatic invigilator assignment.

6.2.1 Outputs

- Merged exam sheet

- Exam grid (Date × Slot)

- Course legend

- Colored, readable Excel format

6.3 faculty.py — Faculty Timetable Generator

This module:

- Reads the generated class timetable.

- Extracts Day × Slot × Room × Course.

- Maps course codes to faculty using CSVs.

- Creates individual sheets for each faculty member.

6.3.1 Outputs

- All faculty timetables

- Index sheet listing all faculty and total class count

7 Troubleshooting

- **Missing CSV columns:** Check for correct headers (case-sensitive).

- **Faculty not assigned:** Ensure Faculty column is not empty.

- **Empty timetable blocks:** Course may have 0 students.

- **Overlapping slots:** Check elective basket definitions.

8 Conclusion

This automation suite significantly reduces manual effort in:

- Class timetable preparation

- Exam planning and room allocation

- Faculty timetable creation

It ensures fairness, accuracy, and consistency across departments.