

Project Report

Author: Pooja Shinde

Address: Navi Mumbai - 400706

Phone No.: 8689801207

Title: Automated School Timetable Generator

Objective:

To create a timetable for a school.

- **Classes:** Jr. Kg, Sr. Kg, 1st
- **Divisions:** A, B, C, D
- **School Start Time:**
 - Jr. Kg, Sr. Kg: 10:15 AM
 - 1st: 8:15 AM
- **School End Time:** 2:15 PM
- **Period Duration:** 30 mins
- **Lunch Break:**
 - Jr. Kg, Sr. Kg: 12:45 PM - 1:15 PM
 - 1st: 9:25 AM - 9:45 AM, 12:45 PM - 1:15 PM
- **Assembly:**
 - Jr. Kg, Sr. Kg: Tuesday, 10:45 AM
 - 1st: Tuesday, 10:45 AM

Final Objective: Create timetables for Jr. Kg, Sr. Kg, and 1st standard for all the divisions.

Tools and Technologies Used:

- **Programming Language:** Python
 - **Libraries:**
 - pandas: For creating and saving timetables in tabular format as CSV files.
 - datetime: For handling and calculating time slots efficiently.
-

Implementation:

The implementation is broken into several steps, ensuring clarity and systematic construction of timetables.

1. Defining Constants

We defined essential constants for timetable creation:

- Classes (Jr. Kg, Sr. Kg, 1st) and their respective divisions.
- Start and end times for the school day.
- Period duration (30 minutes).
- Break schedules for each class.
- Assembly schedule for Tuesday.

2. Generating Time Slots

- **Purpose:** Create structured time intervals for each day.
- **Code Used:**
 - The datetime library was used to calculate start and end times for 30-minute periods.
 - A function, `create_time_slots`, was implemented to generate a list of time slots between the start and end times.

3. Adding Breaks and Assembly

- **Purpose:** Ensure accurate placement of breaks and assembly within the timetable.
- **Code Used:**
 - Conditional checks were added to assign lunch breaks and assembly slots at the specified times for each grade.

4. Assigning Subjects

- **Purpose:** Distribute subjects evenly across time slots, avoiding overlap with breaks and assembly.
- **Code Used:**
 - A round-robin approach was implemented to cycle through the subject list, ensuring all subjects are covered within the school week.

5. Structuring Data into Timetables

- **Purpose:** Organize and structure data for all divisions and grades.
- **Code Used:**
 - A dictionary was created for each grade, with days of the week as keys and lists of scheduled activities as values.

- pandas DataFrames were used to format the data for easy viewing and exporting.

6. Saving Timetables as CSV Files

- **Purpose:** Save generated timetables in a usable format for administrators.
 - **Code Used:**
 - The `to_csv` function from pandas was used to save each grade's timetable as a separate CSV file (e.g., `Jr_Kg_Timetable.csv`).
-

Code Workflow:

1. **Initialization:** Define constants for classes, timings, subjects, and breaks.
 2. **Time Slot Creation:** Use the `create_time_slots` function to generate structured time intervals.
 3. **Timetable Generation:**
 - Loop through each grade, division, and day of the week.
 - Assign breaks, assembly, and subjects to appropriate time slots.
 4. **Save Output:** Convert the structured timetable for each grade into a DataFrame and save it as a CSV file.
-

Outputs:

- Separate timetables were generated for:
 - Jr. Kg (Divisions A, B, C, D)
 - Sr. Kg (Divisions A, B, C, D)
 - 1st (Divisions A, B, C, D)
 - Each timetable includes:
 - Time slots
 - Assigned subjects
 - Break times
 - Assembly and dispersal times
-

Conclusion:

This project successfully automated the creation of school timetables, ensuring structured scheduling and error-free distribution of subjects, breaks, and assembly times. The generated CSV files are ready for use by school administrators and provide a clear format for implementation.