

**Step 1:**

Compare the employees' currently provided holidays to the following reference holiday names and check whether each employee holiday is included.

**Reference Holidays:**

2025-01-01 New Year's Day  
2025-01-20 Martin Luther King Jr. Day  
2025-02-17 Presidents Day  
2025-04-18 Good Friday  
2025-05-26 Memorial Day  
2025-06-19 Juneteenth National Independence Day  
2025-07-04 Independence Day  
2025-09-01 Labor Day  
2025-10-13 Columbus Day  
2025-11-04 Election Day  
2025-11-11 Veterans Day  
2025-11-27 Thanksgiving Day  
2025-11-28 Court Recess (Day after Thanksgiving)  
2025-12-25 Christmas Day  
2025-12-26 Court Recess (Day after Christmas)

Ignore dates and years - match by holiday name only. Day after Christmas won't exist if Christmas doesn't fall on a Thursday:

2026-01-01 New Year's Day  
2026-01-19 Martin Luther King Jr. Day  
2026-02-16 Presidents Day  
2026-04-03 Good Friday  
2026-05-25 Memorial Day  
2026-06-19 Juneteenth National Independence Day  
2026-07-03 Independence Day (observed)  
2026-07-04 Independence Day  
2026-09-07 Labor Day  
2026-10-12 Columbus Day  
2026-11-03 Election Day  
2026-11-11 Veterans Day  
2026-11-26 Thanksgiving Day  
2026-11-27 Court Recess (Day after Thanksgiving)  
2026-12-25 Christmas Day

If any holidays need to be added or removed based on the comparison, proceed to Step 2. If no changes are needed, skip to Step 3.

## Step 2:

1. Open the generate\_holidays.ipynb or refer to code in the **Appendix**
2. To see which NJ Courts holidays are currently defined, run the following cell (adjust the years=[ ] parameter as needed):

```
#Example of Holidays in 2025
nj_holidays = NJCourtsHolidays(years=[2025])
for dates2, name in sorted(nj_holidays.items()):
    print(dates2, name)
```

```
2025-01-01 New Year's Day
2025-01-20 Martin Luther King Jr. Day
2025-02-17 Presidents Day
2025-04-18 Good Friday
2025-05-26 Memorial Day
2025-06-19 Juneteenth National Independence Day
2025-07-04 Independence Day
2025-09-01 Labor Day
2025-10-13 Columbus Day
2025-11-04 Election Day
2025-11-11 Veterans Day
2025-11-27 Thanksgiving Day
2025-11-28 Court Recess (Day after Thanksgiving)
2025-12-25 Christmas Day
2025-12-26 Court Recess (Day after Christmas)
```

3. There is a class NJCourtsHolidays(US). Navigate to def \_populate(self, year).
4. To remove a holiday, refer to the list generated in step 2 for Holiday name.
  - a. ex: removing Labor Day

```
def _populate(self, year):
    super()._populate(year)

    # Remove Lincoln's Birthday 2025-02-12
    self.pop_named("Lincoln's Birthday")
    self.pop_named("Labor Day")
```

Labor Day gets removed:

```
#Example of Holidays in 2025
nj_holidays = NJCourtsHolidays(years=[2025])
for dates2, name in sorted(nj_holidays.items()):
    print(dates2, name)
```

```
2025-01-01 New Year's Day
2025-01-20 Martin Luther King Jr. Day
2025-02-17 Presidents Day
2025-04-18 Good Friday
2025-05-26 Memorial Day
2025-06-19 Juneteenth National Independence Day
2025-07-04 Independence Day
2025-10-13 Columbus Day
2025-11-04 Election Day
2025-11-11 Veterans Day
2025-11-27 Thanksgiving Day
2025-11-28 Court Recess (Day after Thanksgiving)
2025-12-25 Christmas Day
2025-12-26 Court Recess (Day after Christmas)
```

5. To add a holiday
  - a. ex: adding Valentine's Day

```
def _populate(self, year):
    super()._populate(year)

    # Remove Lincoln's Birthday 2025-02-12
    self.pop_named("Lincoln's Birthday")

    # Add Election Day - First Tuesday in November
    self._add_holiday_1st_tue_of_nov("Election Day")
    self._add_holiday_feb_14("Valentine's Day")
```

```
#Example of Holidays in 2025
nj_holidays = NJCourtsHolidays(years=[2025])
for dates2, name in sorted(nj_holidays.items()):
    print(dates2, name)
```

```
2025-01-01 New Year's Day
2025-01-20 Martin Luther King Jr. Day
2025-02-14 Valentine's Day
2025-02-17 Presidents Day
2025-04-18 Good Friday
2025-05-26 Memorial Day
2025-06-19 Juneteenth National Independence Day
2025-07-04 Independence Day
2025-09-01 Labor Day
2025-10-13 Columbus Day
2025-11-04 Election Day
2025-11-11 Veterans Day
2025-11-27 Thanksgiving Day
2025-11-28 Court Recess (Day after Thanksgiving)
2025-12-25 Christmas Day
2025-12-26 Court Recess (Day after Christmas)
```

6. To add a holiday that recurs annually on a particular weekday or weekend within a given month (e.g., the second Monday of March):
  - a. ex:

```
def _populate(self, year):
    super()._populate(year)

    # Remove Lincoln's Birthday 2025-02-12
    self.pop_named("Lincoln's Birthday")

    # Add Election Day - First Tuesday in November
    self._add_holiday_1st_tue_of_nov("Election Day")
    self._add_holiday_2nd_mon_of_mar("Second Monday of March Holiday")
```

```
#Example of Holidays in 2025
nj_holidays = NJCourtsHolidays(years=[2025])
for dates2, name in sorted(nj_holidays.items()):
    print(dates2, name)

2025-01-01 New Year's Day
2025-01-20 Martin Luther King Jr. Day
2025-02-17 Presidents Day
2025-03-18 Second Monday of March Holiday
2025-04-18 Good Friday
2025-05-26 Memorial Day
2025-06-19 Juneteenth National Independence Day
2025-07-04 Independence Day
2025-09-01 Labor Day
2025-10-13 Columbus Day
2025-11-04 Election Day
2025-11-11 Veterans Day
2025-11-27 Thanksgiving Day
2025-11-28 Court Recess (Day after Thanksgiving)
2025-12-25 Christmas Day
2025-12-26 Court Recess (Day after Christmas)
```

### Step 3:

Open the generate\_holidays.ipynb or refer to code in the **Appendix**. To create a new csv file with the specified holidays use the generate\_holidays function. Change starting\_year and ending\_year as needed

ex:

```
df_holidays = generate_holidays(starting_year=2025, ending_year=2040)
```

df_holidays			
	holiday_year	holiday_date	holiday_description
0	2025	01/01/2025	New Year's Day
1	2025	01/20/2025	Martin Luther King Jr. Day
2	2025	02/17/2025	Presidents Day
3	2025	04/18/2025	Good Friday
4	2025	05/26/2025	Memorial Day
...	...	...	...
231	2039	11/11/2039	Veterans Day
232	2039	11/24/2039	Thanksgiving Day
233	2039	11/25/2039	Court Recess (Day after Thanksgiving)
234	2039	12/25/2039	Christmas Day
235	2039	12/26/2039	Christmas Day (observed)

### Appendix

```
from datetime import datetime, date
import holidays
from holidays.countries import US
import pandas as pd
```

```

class NJCourtsHolidays(US):
    """NJ Courts custom holiday calendar.

    Extends UnitedStates holidays with court-specific recess days:
        - Day after Thanksgiving
        - Day after Christmas (only if Dec 25 is Thursday)

    Attributes:
        subdiv (str): Subdivision code ("NJ") by default.
        year (int): Inputted date's year

    Example:
        >>>nj_holidays = NJCourtsHolidays()
        >>>datetime.date(2025, 12, 25) in nj_holidays
        True
        >>>datetime.date(2025, 12, 26) in nj_holidays
        True
        >>>datetime.date(2025, 12, 27) in nj_holidays
        False
        >>>datetime.date(2031, 12, 26) in nj_holidays
        True

    2025 Judiciary Calendar Holidays:
    Jan 1, 20 : New Year's Day, MLK Day
    Feb 17: President's Day
    Apr 18: Good Friday
    May 26: Memorial Day
    Jun 20: Juneteenth
    July 4: Independence Day
    Sept 1: Labor Day
    Oct 13: Columbus Day
    Nov 4, 11, 27: Election Day, Veteran's Day, Thanksgiving
    Dec 25: Christmas Day
    """
    def __init__(self, **kwargs):
        kwargs.setdefault("subdiv", "NJ")
        super().__init__(**kwargs)

    def _populate(self, year):
        super()._populate(year)

        # Remove Lincoln's Birthday 2025-02-12
        self.pop_named("Lincoln's Birthday")

        # Add Election Day - First Tuesday in November
        self._add_holiday_1st_tue_of_nov("Election Day")

        # Day after Thanksgiving (Friday)
        self._add_holiday_1_day_past_4th_thu_of_nov("Court Recess (Day after
    Thanksgiving)")

```

```
# December 26, but ONLY if Dec 25 is a Thursday
christmas = date(year, 12, 25)
if christmas.weekday() == 3: # Thursday = 3
    self._add_holiday_dec_26("Court Recess (Day after Christmas)")
```

```
def generate_holidays(starting_year=2025, ending_year=2036):
```

```
    """
    Generate a DataFrame of holidays defined by `NJCourtsHolidays()`. This function iterates
    through each year in the specified range and compiles all holiday dates and names returned
    by `NJCourtsHolidays()`
    """
```

Args:

starting\_year (int): First year to include (inclusive).  
ending\_year (int): Last year to include (exclusive).

Returns:

pandas.DataFrame: A DataFrame with the following columns:

- holiday\_year (str): The year of the holiday.
- holiday\_date (str): Holiday date formatted as MM/DD/YYYY.
- holiday\_description (str): Name of the holiday.

Example:

```
>>>df_holidays = generate_holidays(starting_year=2025, ending_year=2040)
"""
data = {'holiday_year': [],
        'holiday_date': [],
        'holiday_description':[]}
df = pd.DataFrame(data)

for i in range(starting_year, ending_year):
    nj_holidays = NJCourtsHolidays(years=[i])
    for date_str, name in sorted(nj_holidays.items()):
        year = date_str.strftime('%Y')
        new_date_format = date_str.strftime('%m/%d/%Y')
        df.loc[len(df)] = [year, new_date_format, name]
return df
```

#Example of Holidays in 2025

```
nj_holidays = NJCourtsHolidays(years=[2025])
for dates2, name in sorted(nj_holidays.items()):
    print(dates2, name)
```

```
df_holidays = generate_holidays(starting_year=2025, ending_year=2040)
```

```
df_holidays.to_csv("holidays.csv")
```