SAS INTERVIEW PRESENTATION

Maurice M. Materise

AGENDA

- Prompt
- Tools
- Initial Assumptions
- Solution
- Test Cases
- Example Results
- User Interface
- Challenges
- Questions
- Thanks

PROMPT

A customer is asking for a script / program they can give out to their HR department to help speed up business operations. You have been given the following acceptance criteria:

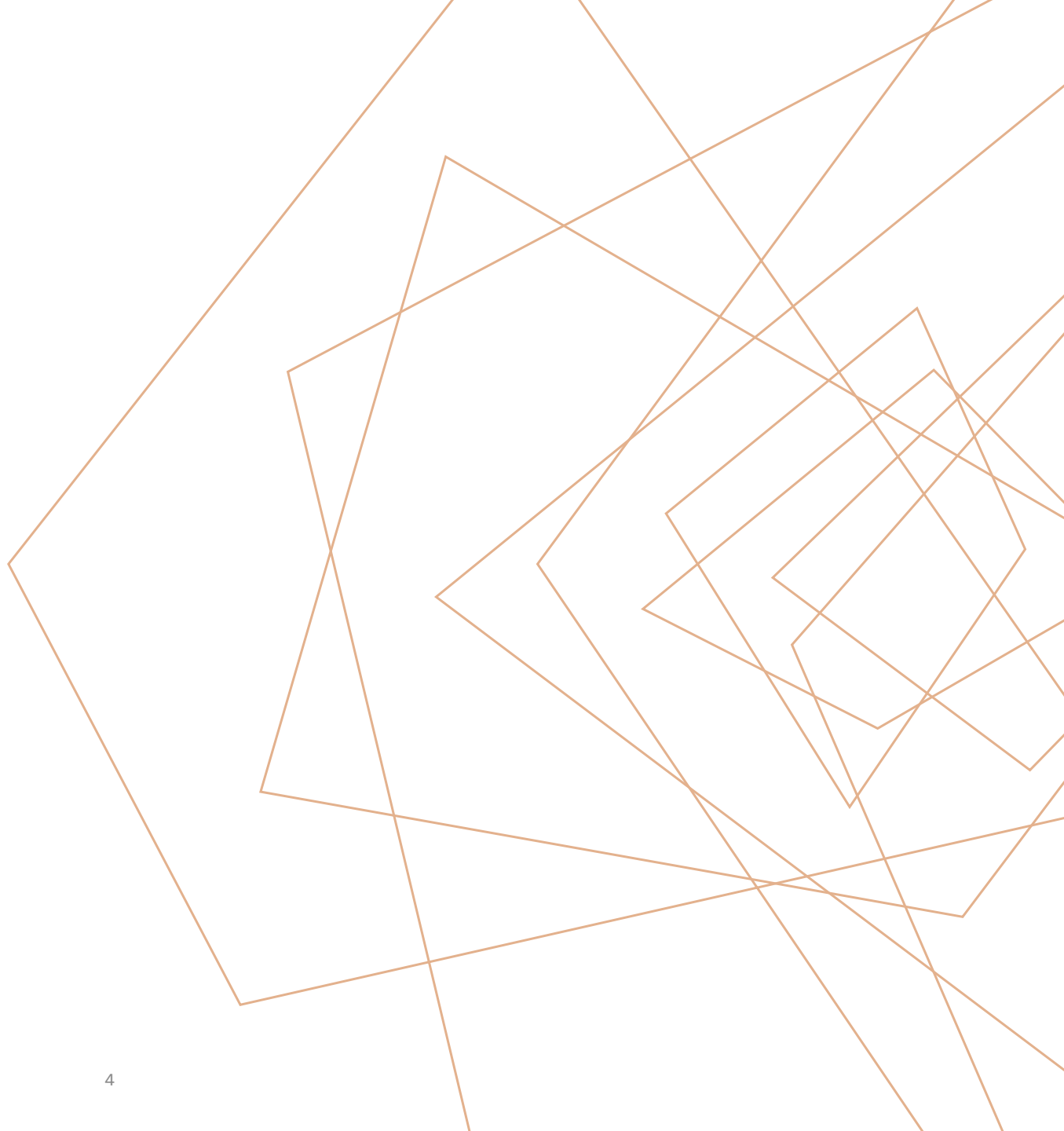
- Given I have an array of employee records that include their birthday
- When I execute this script/program with the array of employees
- Then I am returned an array of employees who's birthday is in the current month

Guidelines:

- Build a runnable script/program that meets the above acceptance criteria
- Use the language and development tools of your choice
- The interface with this script or program is up to you
- Should be tested

TOOLS

- Github
- Python
- Flask
- Javascript
- JQuery
- Bootstrap



INITIAL ASSUMPTIONS

Data Format

The data is in a JSON format with key value pairs for employee details.

Use Case

The object is simply to view which employees have birthdays in a select month. HR would want results in a UI, not from a command line result.

Data is Valid

The data is in valid format. Error checks on the data received not included. I would expect the error checking to occur on data input.

SOLUTION

```python
def get_employee_birthdays(month, employees):
    # initial error checks
    if month is None:
        return 'ERROR: Month input is invalid.'
    elif employees is None:
        return 'ERROR: Employees list input is invalid.'
    elif type(month) != int:
        return 'ERROR: Month input is an invalid type.'
    elif type(employees) != list:
        return 'ERROR: Employees list input is an invalid type.'
    elif len(employees) == 0:
        return 'ERROR: No employees found.'
    elif (month < 1) | (month > 12):
        return 'ERROR: Month input is not an accepted value (1-12).'

    # return list
    bd_list = []
    # loop through list of employee objects
    for e in employees:
        bd = e.get('birthday')
        birth_month = bd.split('/')[0]
        if int(birth_month) == month:
            bd_list.append(e)

    return bd_list
```

Example data:

```python
# a simple list of employee objects
def get_simple_employee_list(self):
    return [
        {
            'id':1,
            'first_name': 'Maurice',
            'last_name': 'Materise',
            'birthday': '5/16/1993'
        },
        {
            'id':2,
            'first_name': 'Larkin',
            'last_name': 'Materise',
            'birthday': '7/4/2019'
        },
        {
            'id':3,
            'first_name': 'Smalls',
            'last_name': 'Materise',
            'birthday': '3/15/2013'
        },
        {
            'id':4,
            'first_name': 'John',
            'last_name': 'Smith',
            'birthday': '5/1/1982'
        },
        {
            'id':5,
            'first_name': 'Jane',
            'last_name': 'Doe',
            'birthday': '3/3/1989'
        },
    ]
```

Code: https://github.com/materimm/SAS/blob/main/employees.py

TEST CASES

1. Basic success
2. Null month
3. Null employee list
4. Bad month type
5. Bad employee type
6. Empty employee list
7. Negative month value

Test case library: unittest

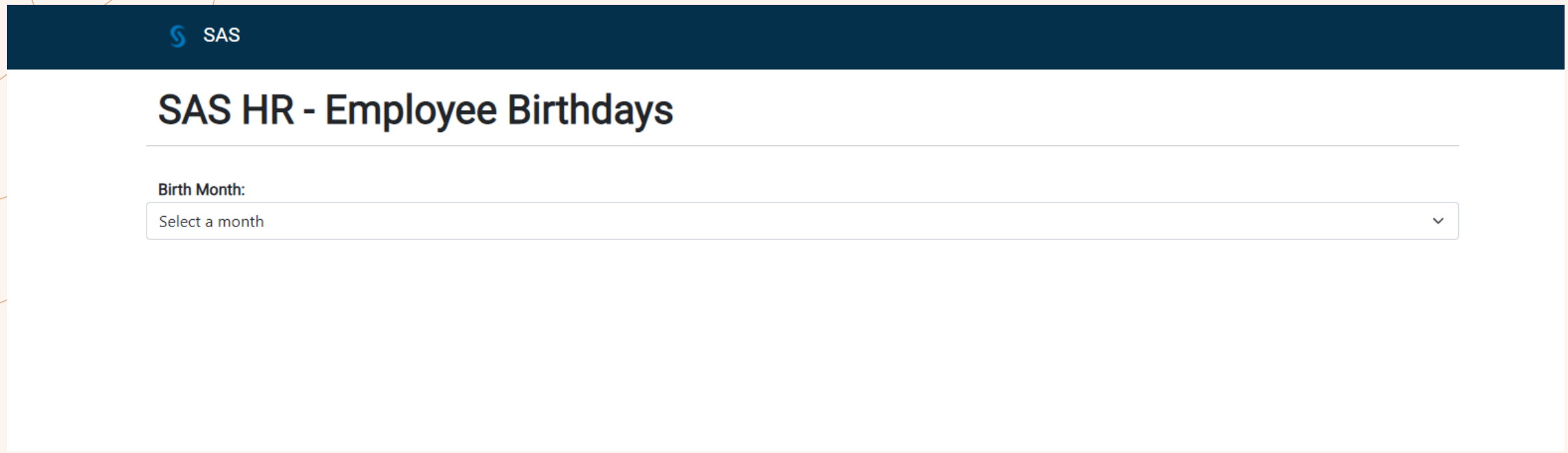
```python
# unit test case class
class TestEmployeeBirthdays(unittest.TestCase):
    # a simple list of employee objects
    def get_simple_employee_list(self):

    # test the cupcake success case
    def test_basic_success(self):
        e = self.get_simple_employee_list()
        actual = get_employee_birthdays(5, e)
        expected = [{
                'id':1,
                'first_name': 'Maurice',
                'last_name': 'Materise',
                'birthday': '5/16/1993'
            },
            {
                'id':4,
                'first_name': 'John',
                'last_name': 'Smith',
                'birthday': '5/1/1982'
            }]
        self.assertEqual(actual, expected)


    # test for when month is none
    def test_null_month(self):
        e = self.get_simple_employee_list()
        actual = get_employee_birthdays(None, e)
        expected = 'ERROR: Month input is invalid.'
        self.assertEqual(actual, expected)


    # test for when the employee list is none
    def test_null_employees(self):
        actual = get_employee_birthdays(5, None)
        expected = 'ERROR: Employees list input is invalid.'
        self.assertEqual(actual, expected)
```

```python
# test for bad type
def test_bad_month_type(self):
    e = self.get_simple_employee_list()
    actual = get_employee_birthdays('5', e)
    expected = 'ERROR: Month input is an invalid type.'
    self.assertEqual(actual, expected)


# test for bad type
def test_bad_employee_type(self):
    actual = get_employee_birthdays(5, 5)
    expected = 'ERROR: Employees list input is an invalid type.'
    self.assertEqual(actual, expected)


# test for an empty list
def test_empty_list(self):
    actual = get_employee_birthdays(5, [])
    expected = 'ERROR: No employees found.'
    self.assertEqual(actual, expected)

# test for negative month
def test_negative_month(self):
    e = self.get_simple_employee_list()
    actual = get_employee_birthdays(-5, e)
    expected = 'ERROR: Month input is not an accepted value (1-12).'
    self.assertEqual(actual, expected)
```

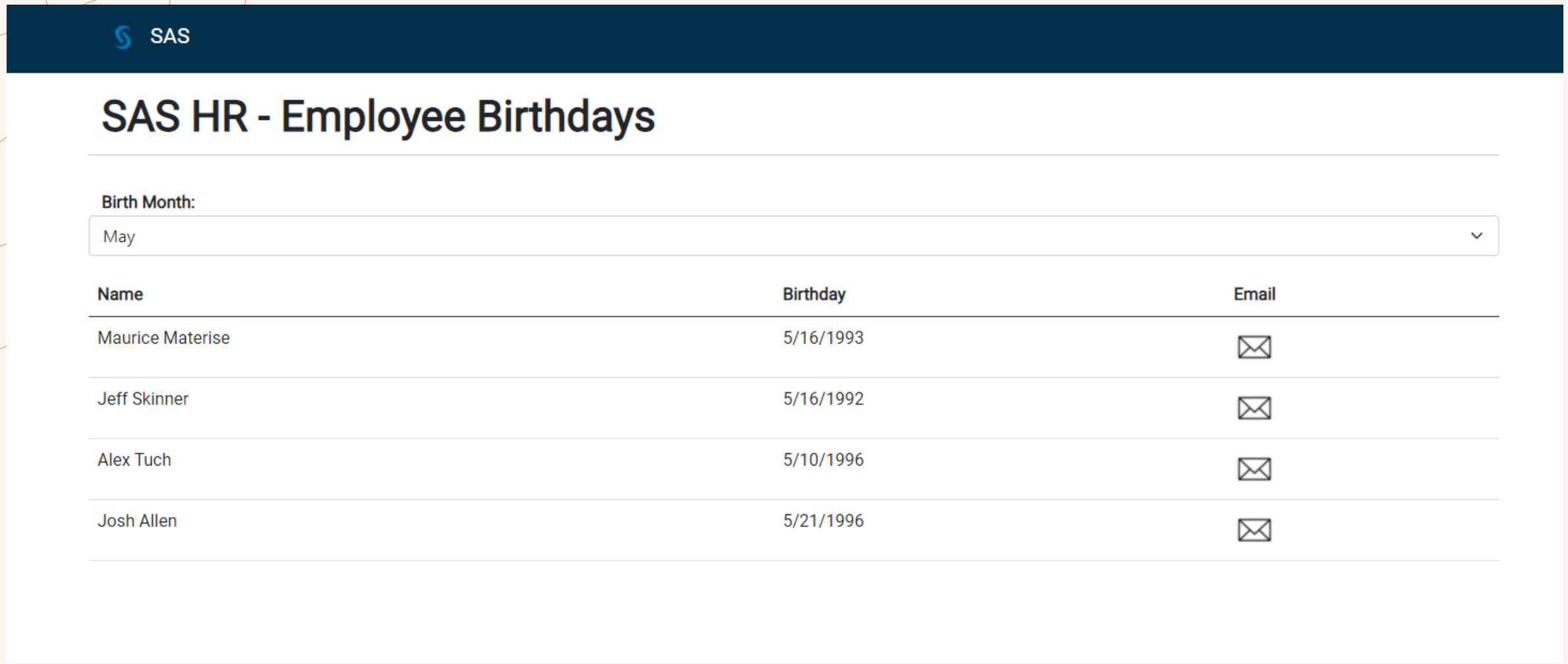
EXAMPLE RESULTS

```
C:\Users\MoreyMATERISE\Documents\sas\SAS>python employees.py
[{'id': 1, 'first_name': 'Maurice', 'last_name': 'Materise', 'birthday': '5/16/1993'}, {'id': 4, 'first_name': 'John', 'last_name': 'Smith', 'birthday': '5/1/1982'}]

C:\Users\MoreyMATERISE\Documents\sas\SAS>python -m unittest employees.py
.......
----------------------------------------------------------------------
Ran 7 tests in 0.001s

OK

C:\Users\MoreyMATERISE\Documents\sas\SAS>
```

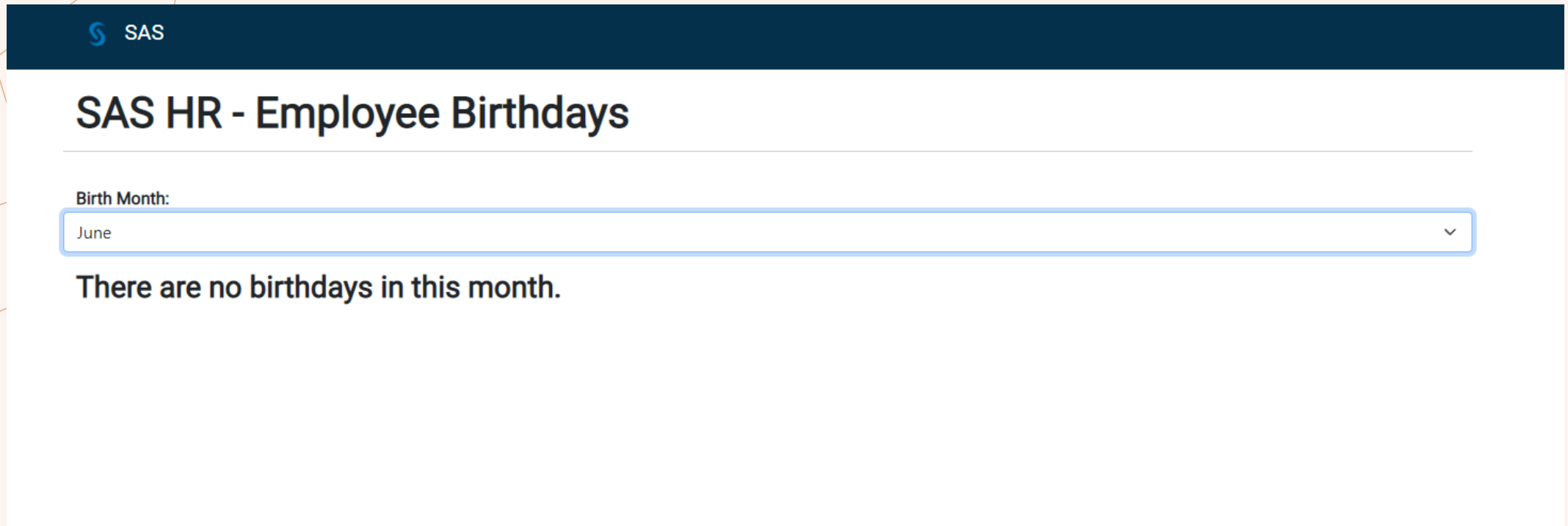
USER INTERFACE



USER INTERFACE



USER INTERFACE



CHALLENGES

USER PROBLEM

USE CASE

What is the problem to solve?

What is the use case for the solution?

Lack of stakeholder feedback

DATA

Where is the data?

Data format

Personal Information access

ERRORS TO CHECK

Bad keys, wrong birthday format

Database injection from UI

QUESTIONS

THANK YOU

Maurice M. Materise

716-796-5849

mmaterise@gmail.com

Full Code Base: https://github.com/materimm/SAS

LinkedIn: https://www.linkedin.com/in/maurice-materise-72301575/