

|  |
| --- |
| ***Klein Dining***  ***Employee Management System*** |

|  |
| --- |
| **IS213 Enterprise Solution Development**  *G4-T7*  **Assignment**  Jazreel Tho  Lee Cheng Leng  Ng Jun Hong  Siew Wei Lun  Soh Wei Ming |

### Introduction

Our application encompass 3 business scenarios that a restaurant needs to manage its employees, comprising of 11 user scenarios.

**Business Scenarios**

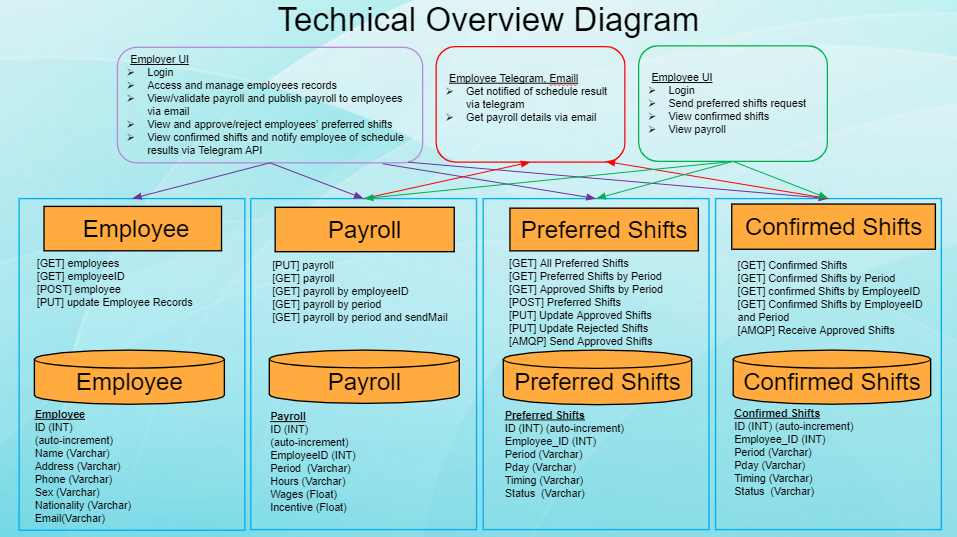
1. Payroll
2. Shift Scheduling

**User Scenarios**

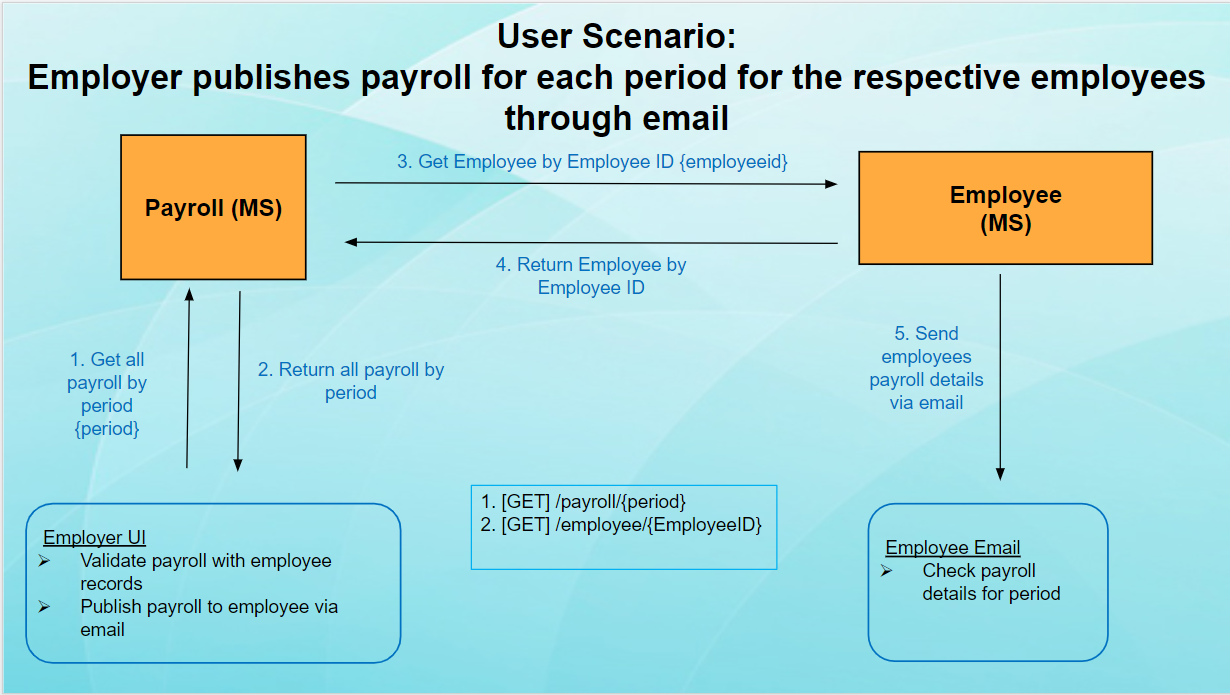
1. Payroll Microservice
   1. Employer views payroll details by employeeID or period
   2. Employer publishes payroll for each period for the respective employees through email
2. Preferred Shift Microservice
   1. Employee select their preferred shifts for the upcoming period
   2. Employer approve/reject employees’ preferred shift to determine confirmed schedule
   3. Employer view confirmed shifts in final schedule
3. Confirmed Shifts Microservice
   1. Employer publishes confirmed shifts
   2. Employer view confirmed shift schedule

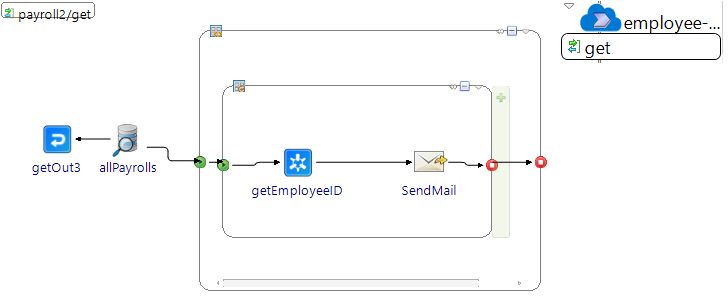
\*Note: User Scenarios pertaining to the Employee Microservice were left out due to page limit.

### Technical Overview Diagram

**

#### **User Scenario 1**: Employer publishes payroll for each period for the respective employees through email





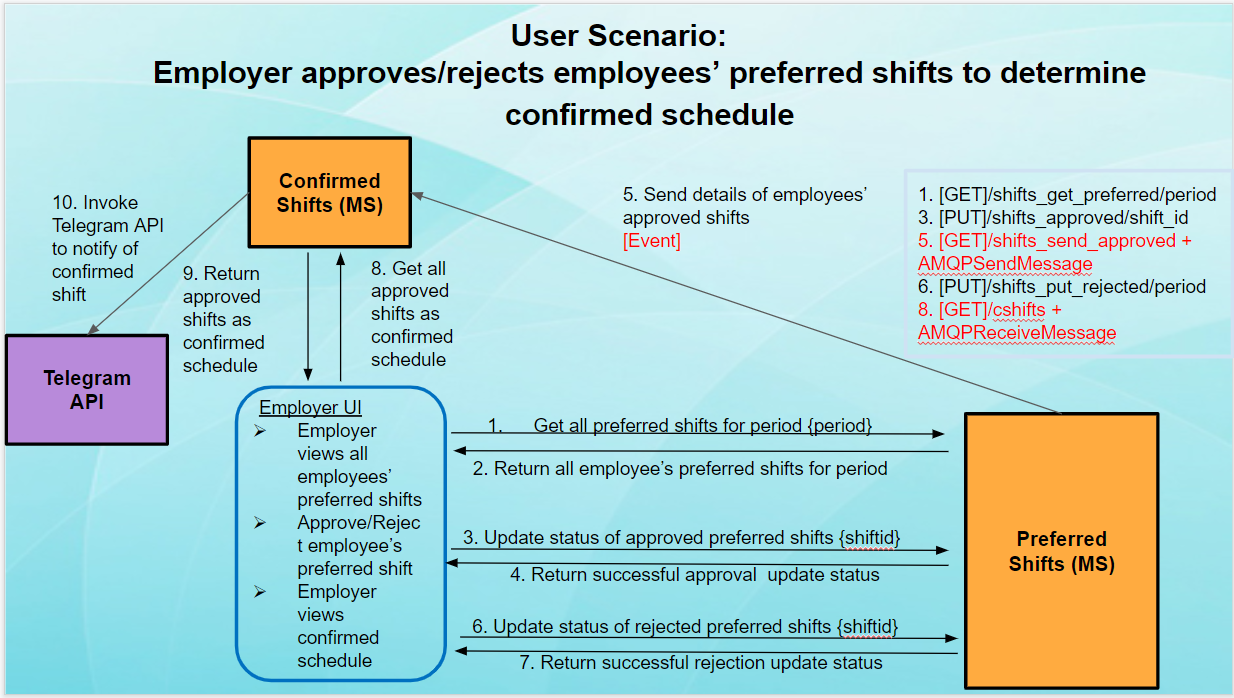
The Employer will view employee payrolls by period. For this, the Employer UI will make a **GET** request via **JDBC Query** to the Payroll Microservice for the list of employee payrolls in the specified period. The Payroll Microservice will return the information in a table to the Employer UI. For each payroll record retrieved, the Payroll Microservice will make a **synchronous call** to the Employee Microservice to retrieve the employee email belonging to that EmployeeID. The Payroll Microservice will then send an email of their payroll to each employee through the **Send Mail activity**.

*\*Assumption: Actual Hours Worked is tabulated by a time in/out system outside of the scenario*

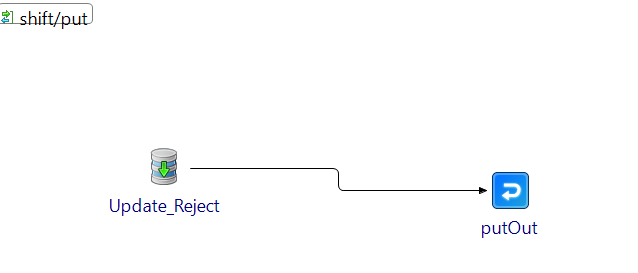
#### **User Scenario 2**: Employees select their preferred shifts for the upcoming period

Employees will submit their preferred shift timings through the Employee UI. The Employee UI will then invoke the Preferred Shifts microservice through the **POST** method via **JDBC Update** to insert the information into the Preferred Shifts database. The Employee UI will then display the submitted preferred shifts to the users.

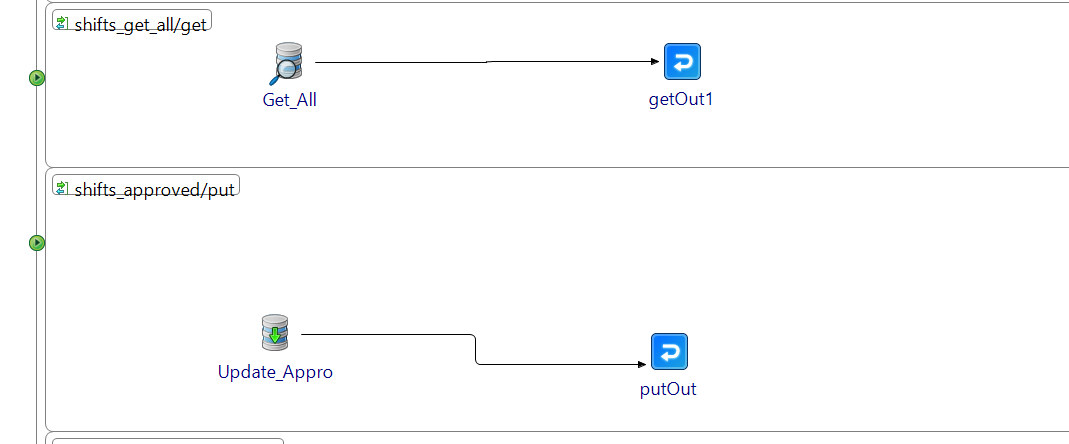
#### **User Scenario 3**: Employer approves/rejects employee’s preferred shifts to determine confirmed schedule

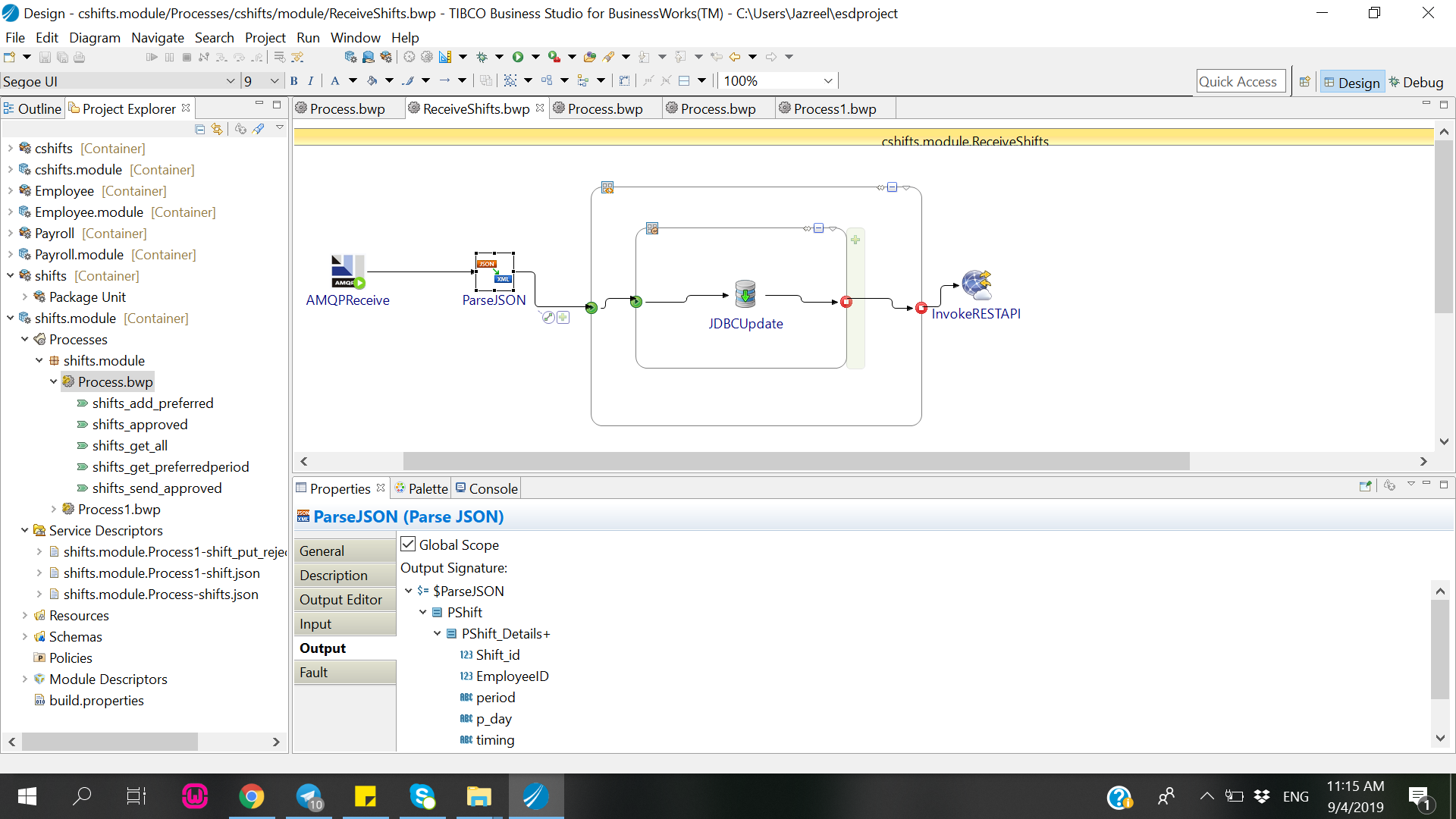


3)

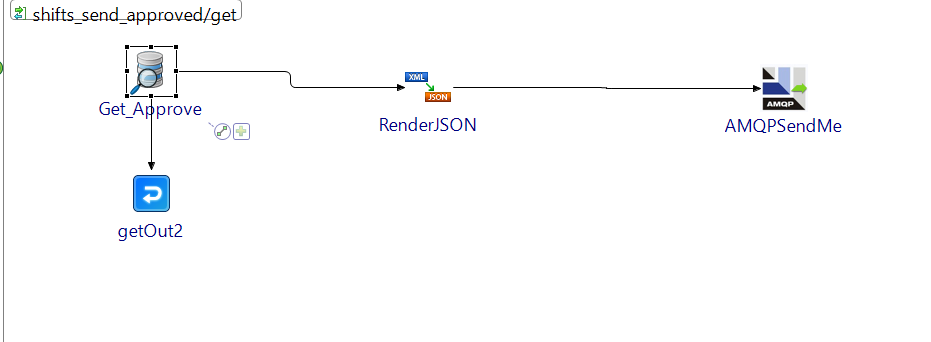


1) and 2)



5) and 6)

4)

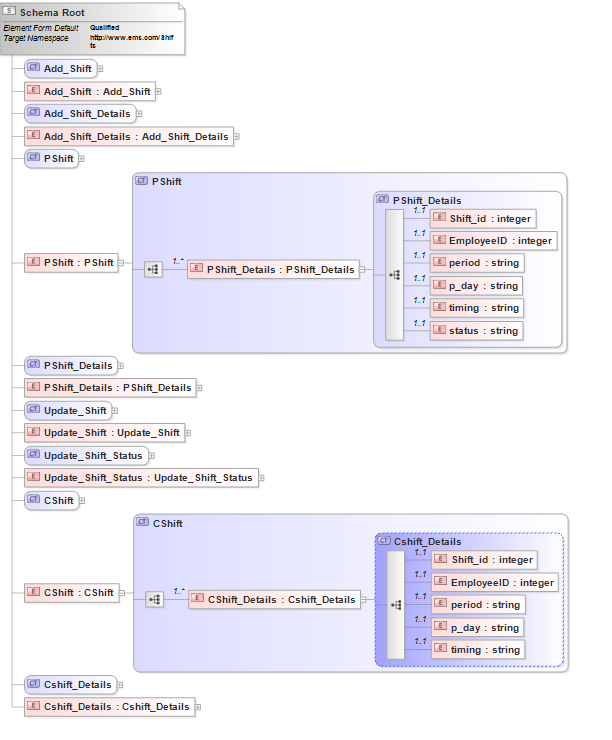
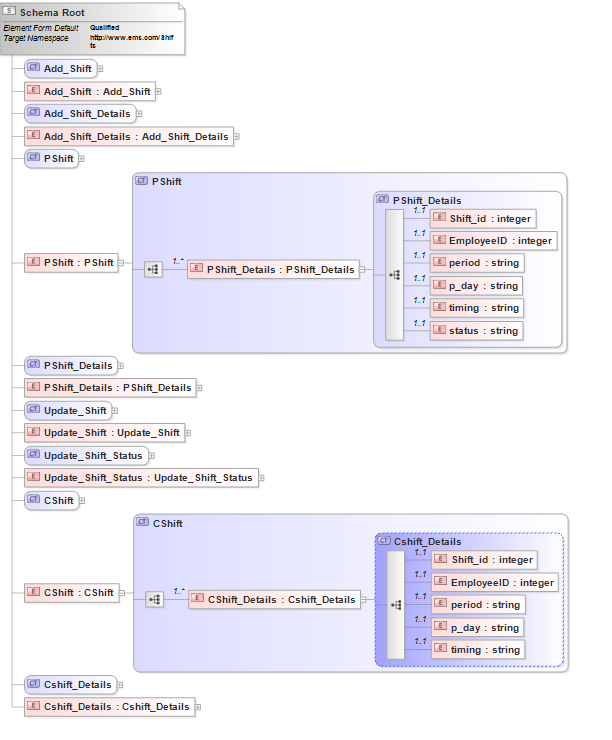


1. The Employer UI invokes Preferred Shifts microservice using **GET** by period via **JDBC Query**. Employer will choose the respective employees’ shifts which he wishes to approve by ticking the checkboxes in its user interface.
2. Upon clicking the submit button, the user interface will invoke the Preferred Shifts microservice via **PUT** by shift\_idvia **JDBC Update** to update the respective shifts’ status to “Approved” based on the shift\_id.
3. Another **PUT** is used to change the status of the remaining shifts in the period to “rejected” in the Preferred Shifts Database.
4. The Employer UI then makes makes **JDBC Query** to the Preferred Shifts Database to retrieve all approved shifts in the period, and sends these shifts via **AMQPSendMessage** using **RabbitMQ** broker to the ‘pshift’ queue.
5. Once ready, the Confirmed Shift microservice receives the approved shifts via **AMQPReceiveMessage** and updates the Confirmed Shifts database with the approved shifts via **JDBC Update**
6. Confirmed Shifts microservice calls the **external Telegram API** to notify the employees via Telegram group chat that a new schedule has been posted

### Web Services

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Service | Operation | Description | Input | Output |
| Payroll service | *Get payrolls by period* | *Retrieve all payroll records in Payroll Service base on selected period*  *GET/payroll\_by\_period* | *period* | *Array of payrolls* |
| *Get payrolls by employeeid* | *Retrieve all payroll records in Payroll Service base on selected employeeID*  *GET/payroll\_by\_employeeid* | *eid* | *Array of payrolls* |
| *Put payroll records* | *Insert new payroll record in Payroll Service database*  *PUT/payroll* | *Id, eid, period, hours, wages, incentive* | *Payroll* |
| *Get all payroll records* | *Retrieve all employee records in the Payroll Service database*  *GET/payroll1* | *nil* | *Array of payrolls* |
| *Email employees’ payroll records* | 1. *Payroll Microservice make a synchronous call to employee microservice to retrieve employees’ emails from employee records based on Payroll EmployeeId*   *GET/payroll2*   1. *Payroll Microservice send out emails to the retrieved employee records their payroll* | 1. *Period* 2. *from, to, message* | *1)Payroll*  *2)Payroll message (e.g. Your payroll for 20190101 is $300)l* |
| Simple Mail Transfer Protocol (SMTP) | *N/A* | *Simple Mail Transfer Protocol, handles the procedure of email trade ,where employer publishes payroll to employees*  *SMTP method,* [*https://www.smtp2go.com/*](https://www.smtp2go.com/) |  | *nil* |
| Preferred Shifts service | *Post preferred shifts* | *Employees select preferred shifts for upcoming period*  *POST/shifts\_add\_preferred* | *employeeID,*  *period, p\_day, timing, status* | *Array of preferred shifts* |
| *Get all preferred shifts* | *Retrieve all employees’ preferred shifts in Preferred Shifts database*  *GET/shifts\_get\_all* | *nil* | *Array of preferred shifts* |
| *Get preferred shifts by period* | *Employer retrieve all employees’ preferred shifts for each period*  *GET/shifts\_get\_preferred/{period}* | *period* | *Array of preferred shifts* |
| *Put approve preferred shifts* | *Employer approve employees preferred shifts*  *PUT/shift\_approved* | *shift\_id, employeeID, period, pday, timing, status* | *nil (response with status code)* |
| *Put reject preferred shifts* | *Employer reject employees preferred shifts*  *PUT/shift\_put\_rejected/{period}* | *shift\_id, employeeID, period, pday, timing, status* | *nil (response with status code)* |
| *Send approve shifts to confirmed shifts service* | *Preferred shift utilize AMQPSendMessage to send approve shifts to confirmed service*  *GET/shifts\_send\_approved/{period}* | *nil* | *Array of approved shifts* |
| Confirmed Shifts service | *Get all confirmed shifts* | *Retrieve all confirmed shifts in the confirmed shifts service database*  *GET/cshifts* | *nil* | *Array of confirmed shifts* |
| *Get all confirmed shifts by employeeID* | *Retrieve the confirmed shifts of a specific employee*  *GET/cshifts\_get\_eid/{eid}* | *eid* | *Array of confirmed shifts* |
| *Get all confirmed shifts by period* | *Retrieve the confirmed shifts base on selected period*  */cshifts\_get\_period/{period}* | *period* | *Array of confirmed shifts* |
| *Get all confirmed shifts by period and employeeID* | *Retrieve the confirmed shifts of a specific employee base on selected period*  */cshifts\_get\_periodeid/{period}/{eid}* | *eid ,*  *period* | *Array of confirmed shifts* |
| *Telegram API* | *N/a* | *Employer retrieves confirmed schedule for all employees for the respective period and invoke the telegram API* | *nil* | *Telegram Bot sends a notification to the employee telegram group chat* |

### Usage of XML schema

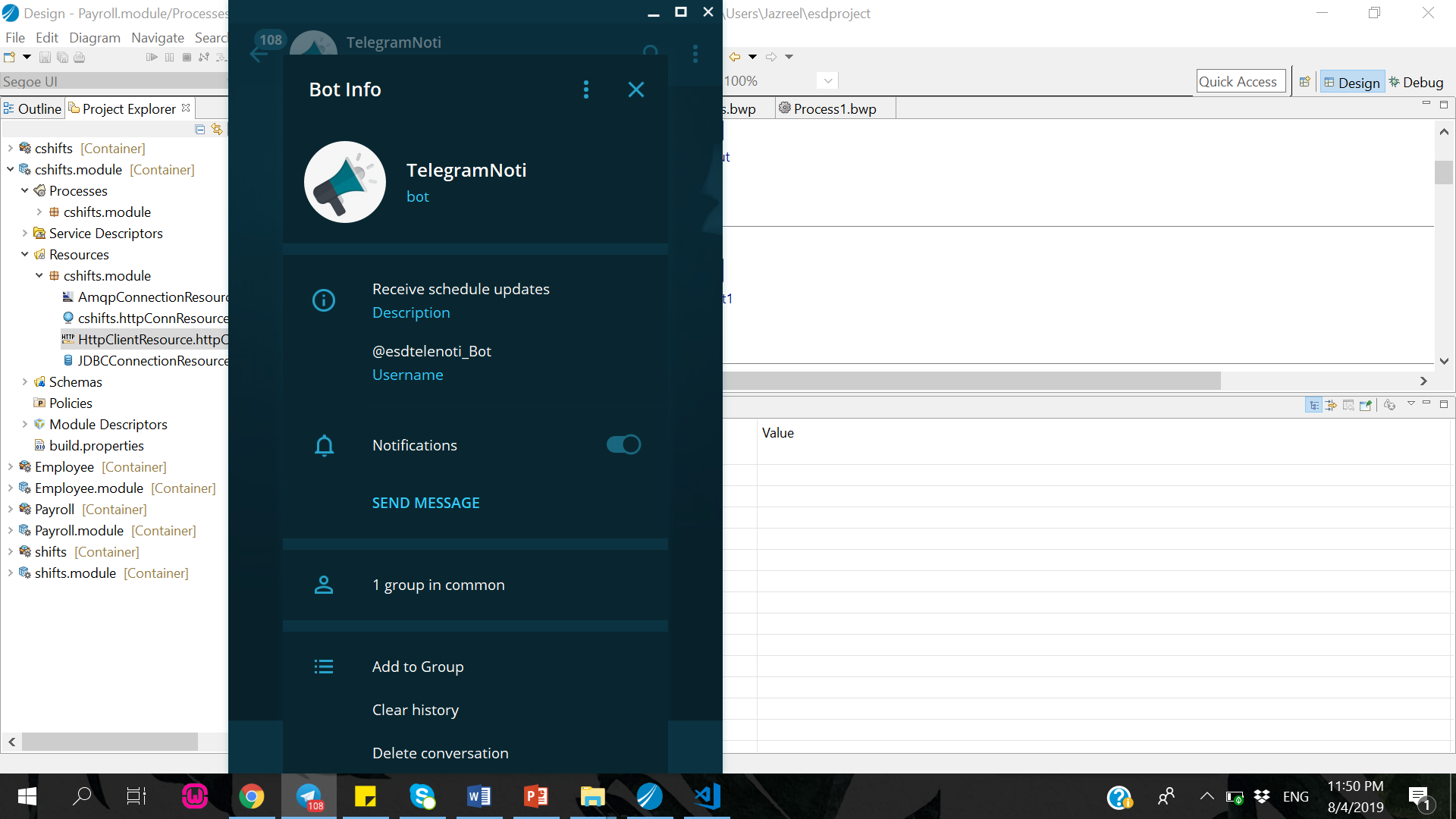
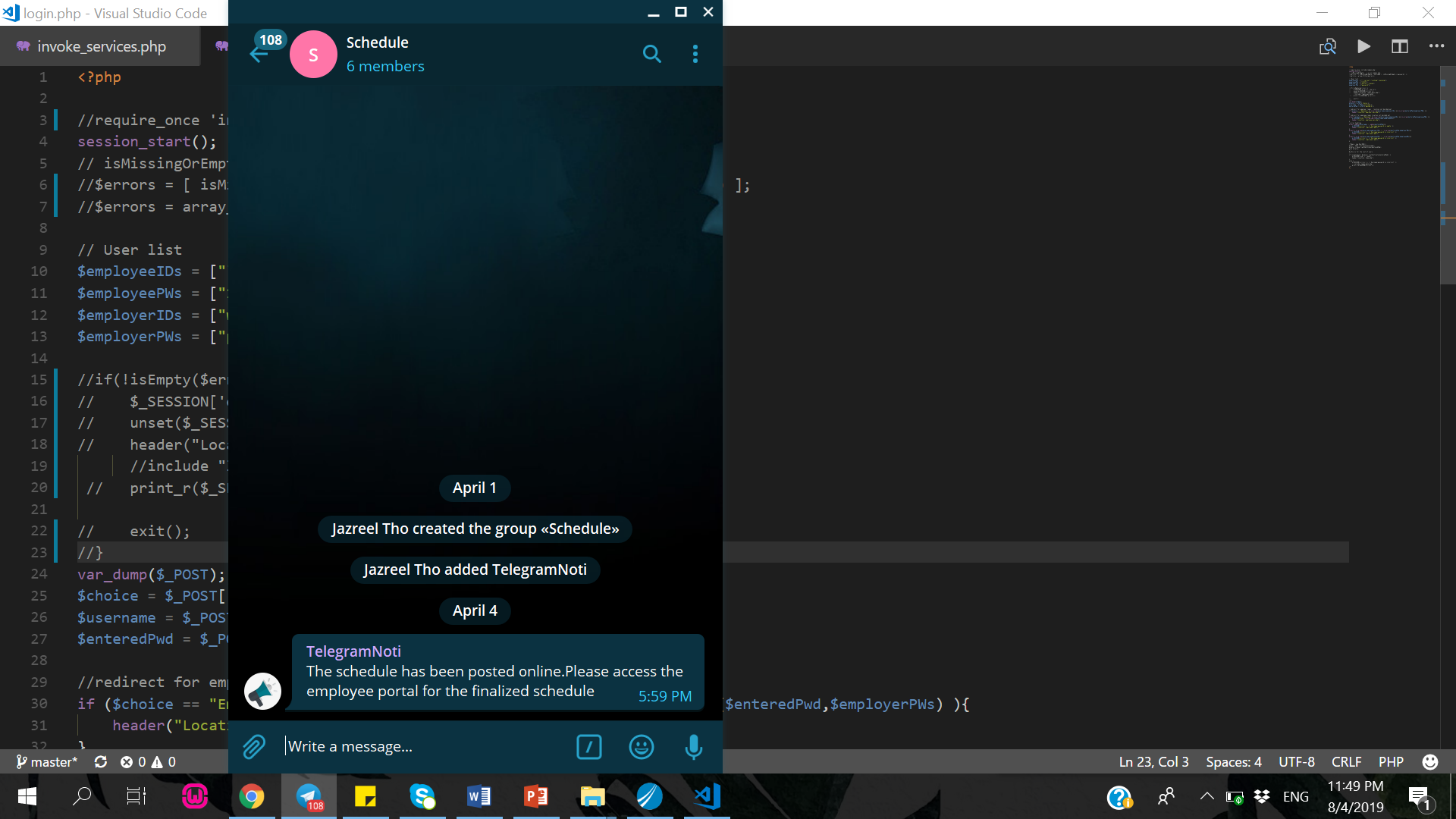


One unique usage of XML Schema was in our shifts.xsd. Within this single XSD file, we created P Shift and C Shift elements to cater to both Preferred Shifts microservice and Confirmed Shifts microservice respectively. This is to cater to the differences in their structure.

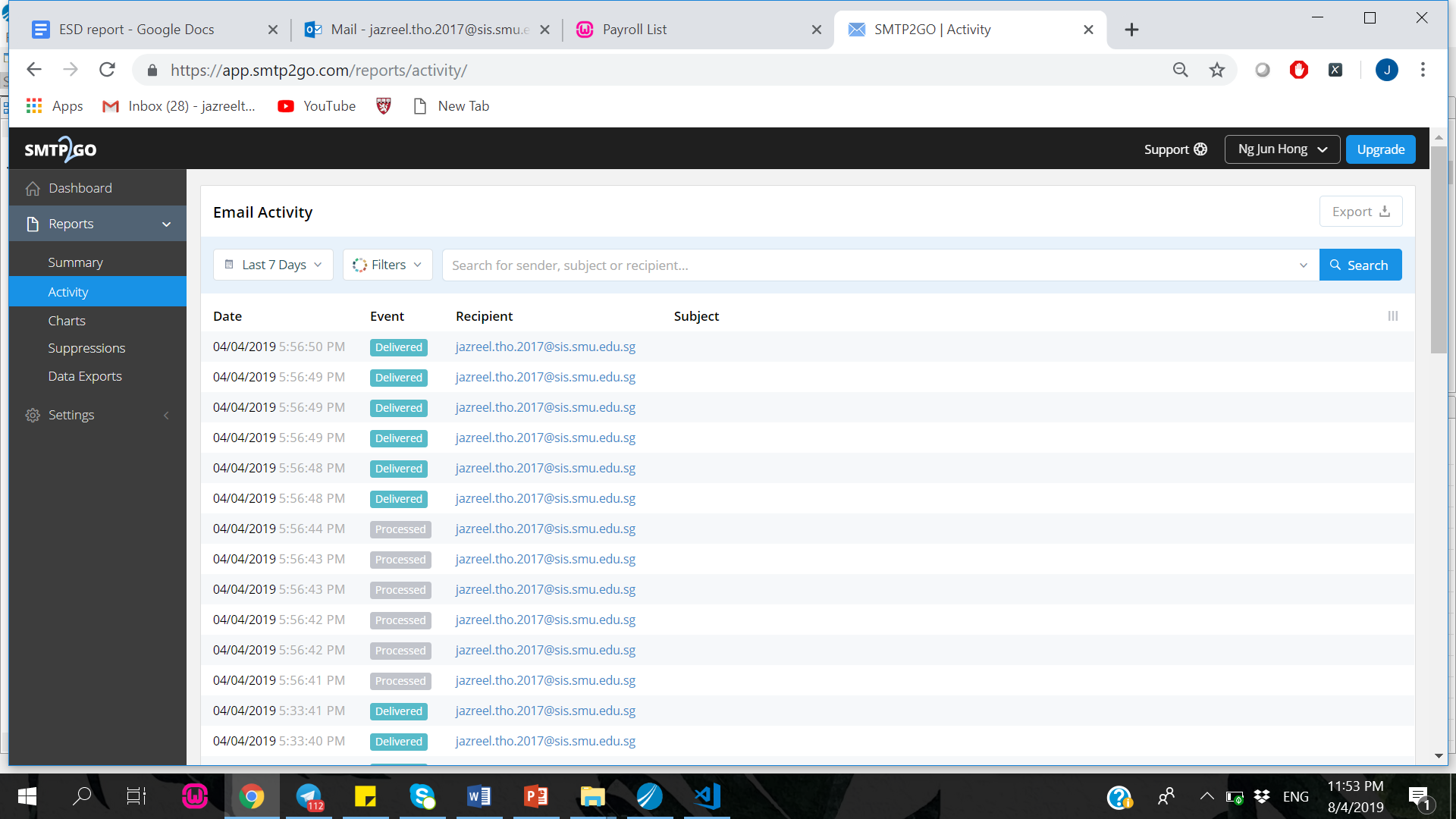
### Graphical User Interface

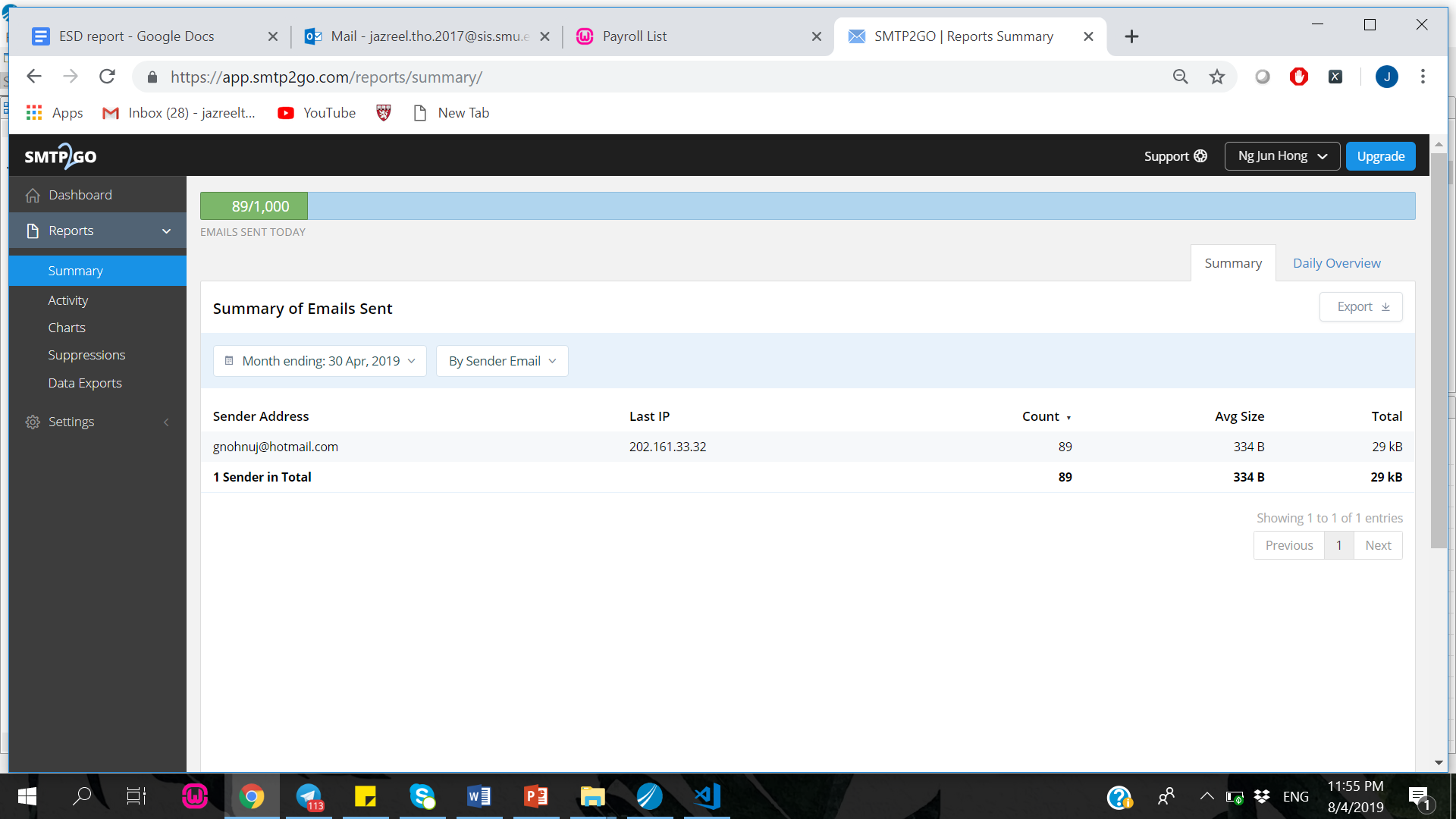
We designed our user interface using PHP. In addition, we made use of CSS and bootstrap functions to beautify our web pages. Some examples would be using of navigation and tab buttons across pages. CSS functions are used to design and insert colors to our tables across various pages. To invoke our services created in TIBCO, we made use of HTTP GET and POST requests using **cURL**.

### Beyond the Labs

**

We created a **Telegram Bot** and added it into the group chat that consists of all employees. Once the employer confirms the shift schedule, the **Telegram API** will be invoked and it will send a notification in the group chat, notifying the employees that employees are able to check their schedule online.





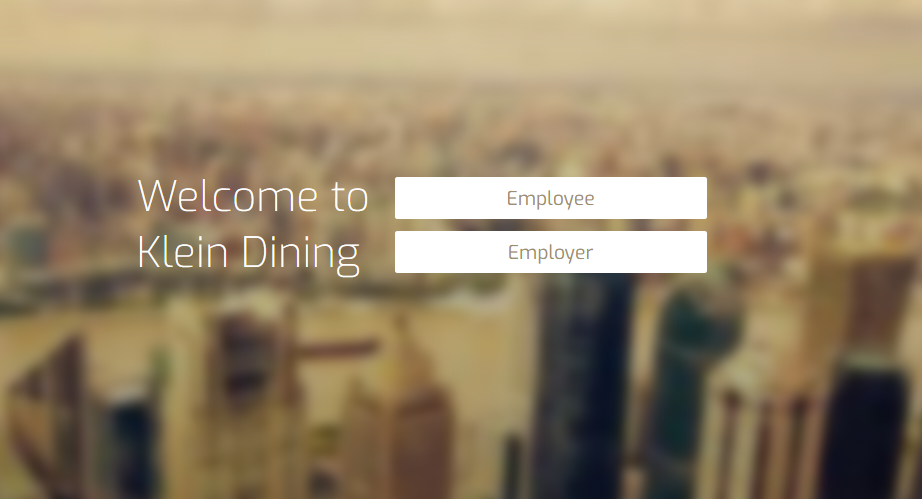
Another component that we had was **SendMail**. Upon checking of the payroll details of the employees, the employer will click send email on the portal and an email will be sent out to all employees who worked for that respective period. We also utilised **for loop** and **local transaction** for this scenario, as well as for other scenarios. We used smtp2go for the stmp connection. It also enables users to track the number to emails sent via the process and who the email has been sent to as well.

**For loop** allows us to loop through every record in the payroll database. For every record in the payroll database, it will be mapped to the employee database by Employee ID to obtain the respective employee’s email as we needed the email and payroll details of the employee to invoke the sendMail node.

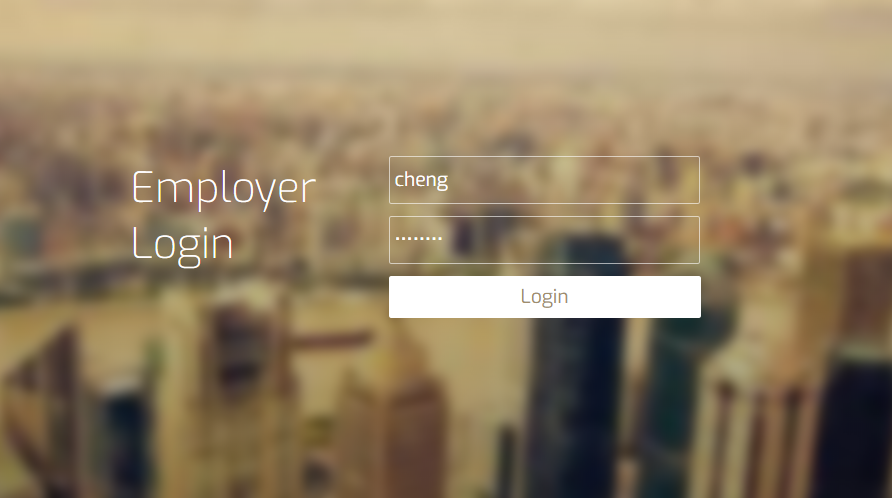
As for the **local transaction**, it will stop the process if there is any errors and it will rollback to the last node that ran successfully. This prevent any errors from being passed by the microservice.

We also used **Render JSON** to convert the format of the data from XML to JSON and **Parse JSON** to convert the output of the data from JSON format to XML in the shift scenario.

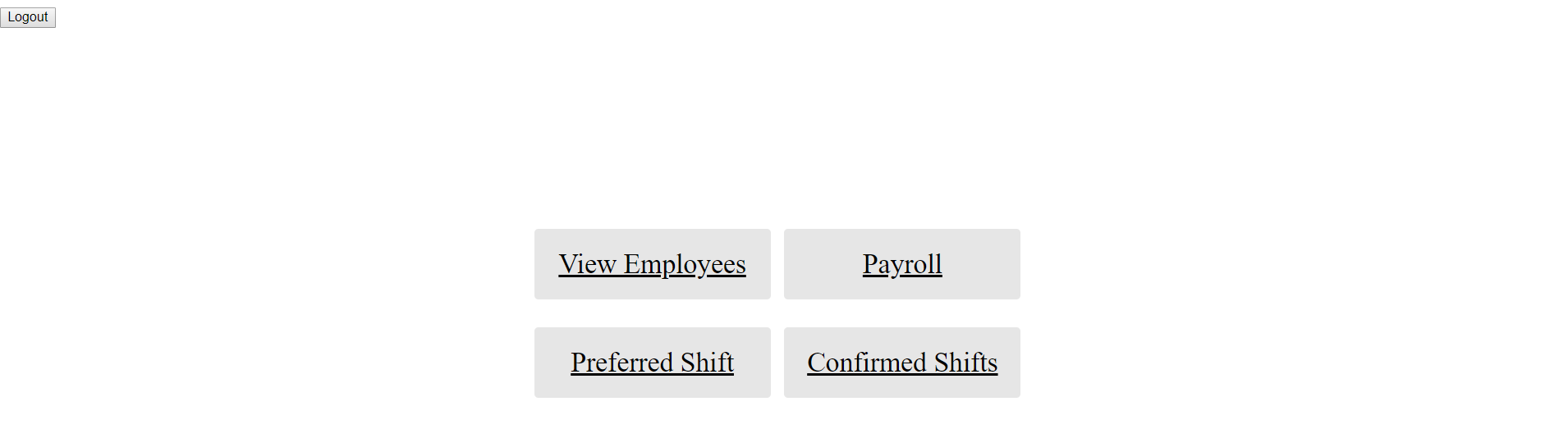
### **Scenario Walkthrough**

**

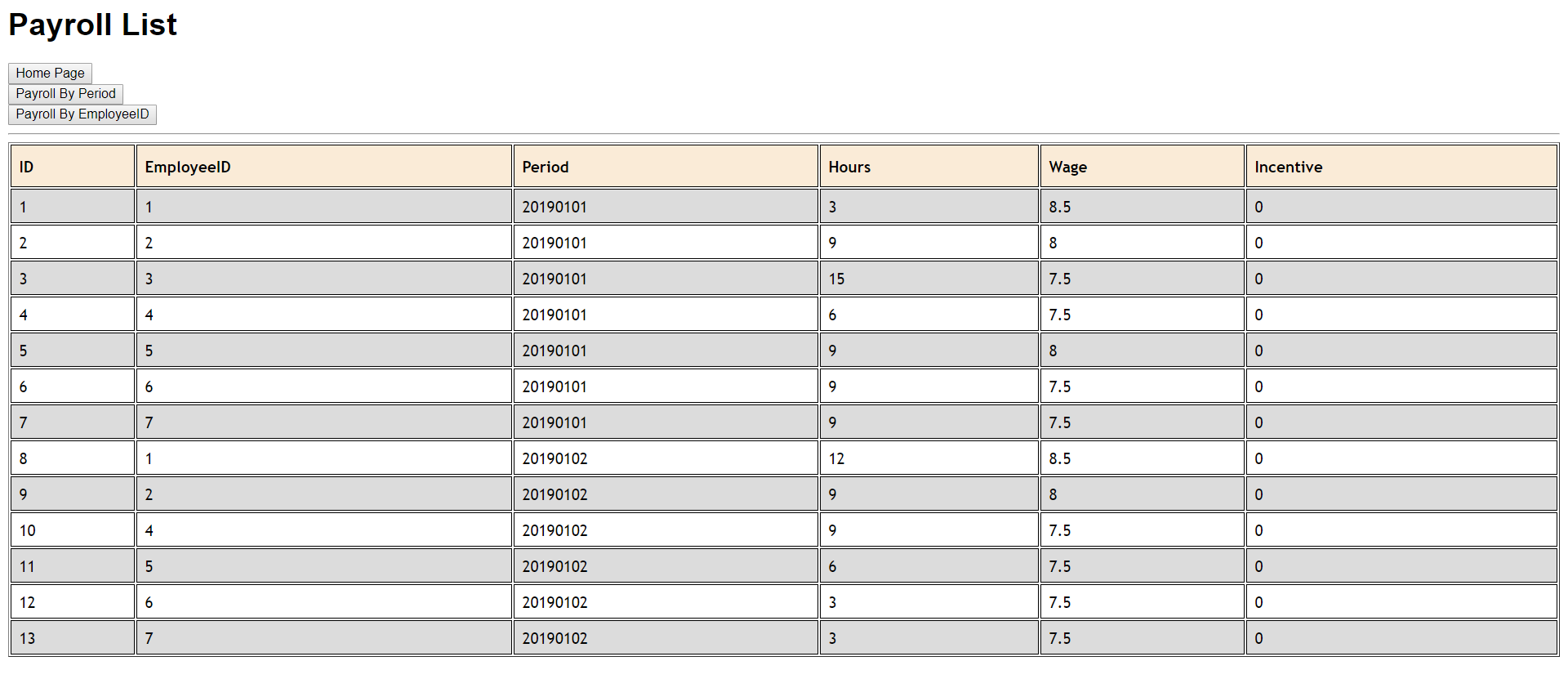
1) Navigate to login\_main.php and click on the “Employer” button to access employer login user interface.

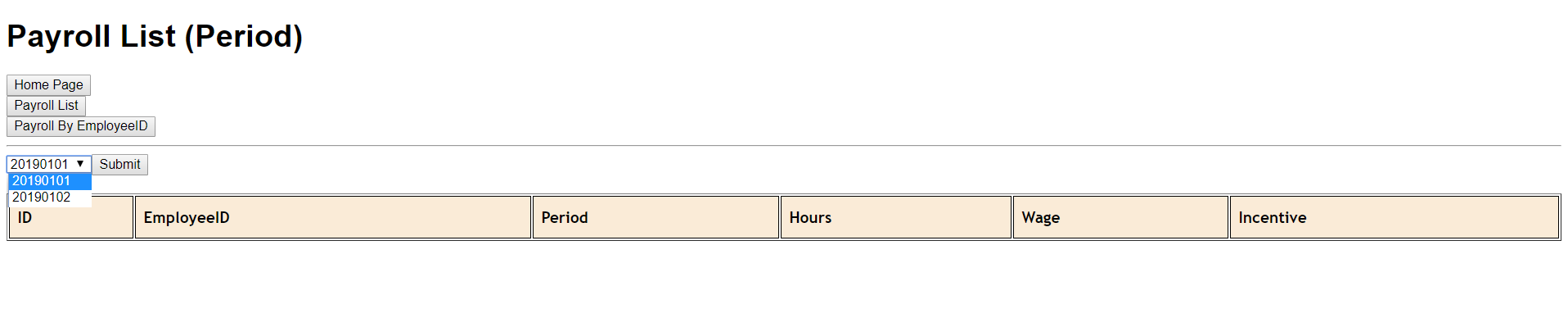


2) Login to Employer user interface with username: cheng and password: password.

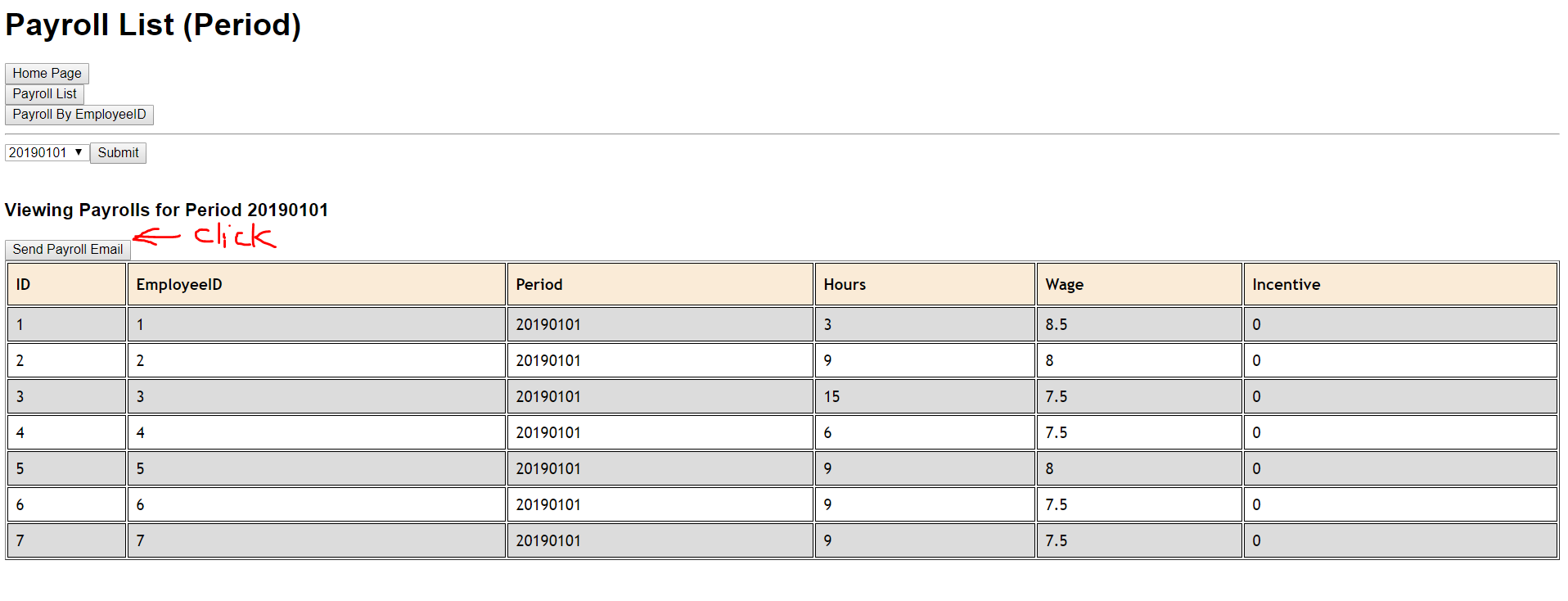
**

3) You will be directed to this home page as shown above. Click “Payroll” button to access employer payroll function.

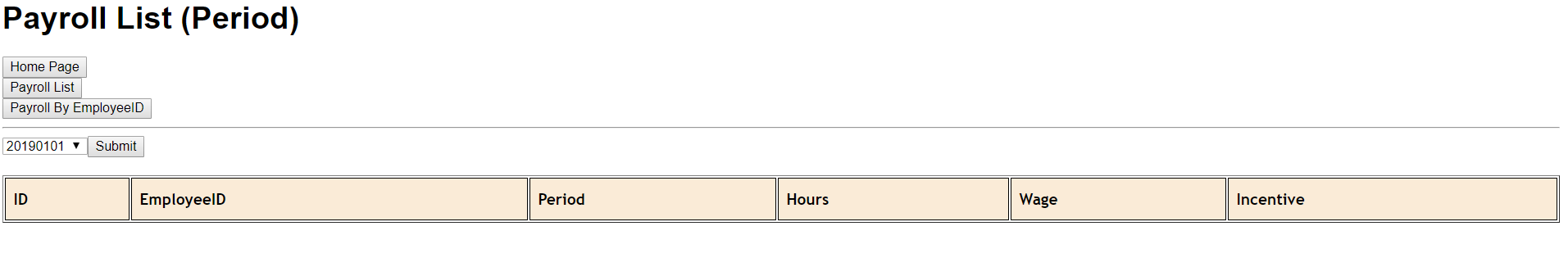
**

4) Upon reaching this page, click on the “Payroll by Period” button on the top left of the screen to access payroll items in the specific period.

5) Choose the period “20190101” in the dropdown list as shown above and click the “submit” button. “20190101” refers to the first week of January 2019. Payroll items are tabulated at the end of each week.

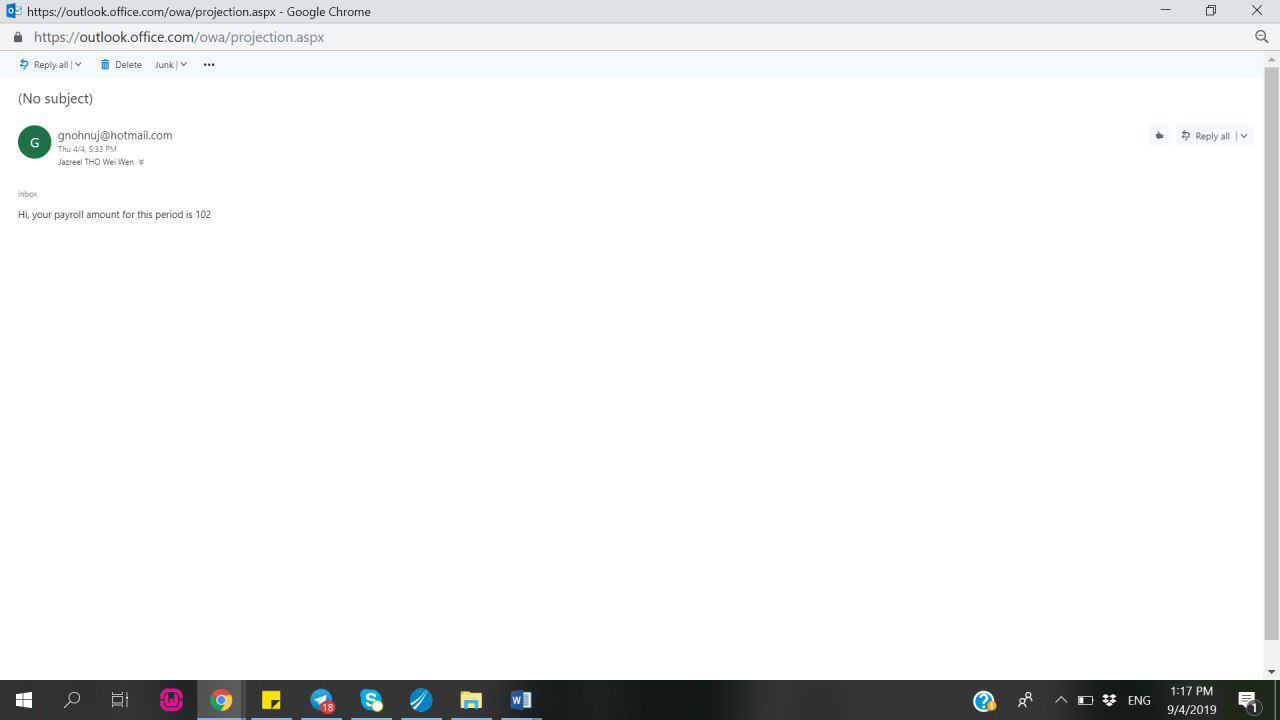


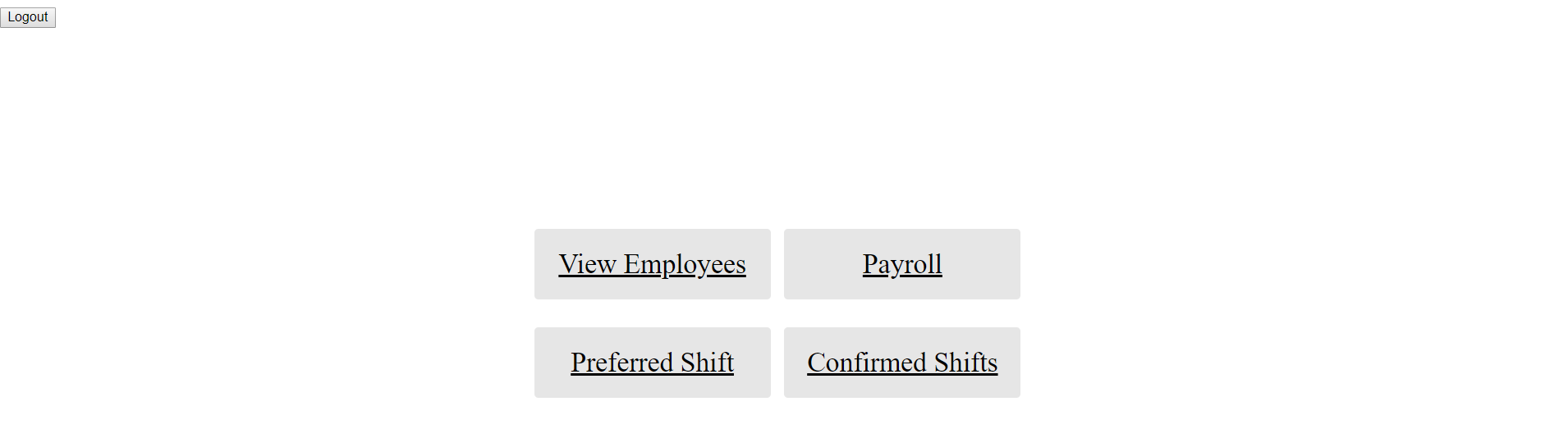
1. On this page, you would be able to view the respective payroll items in the selected period “20190101”. Click on the “Send Payroll Email” button to send out notification emails to the respective employees.



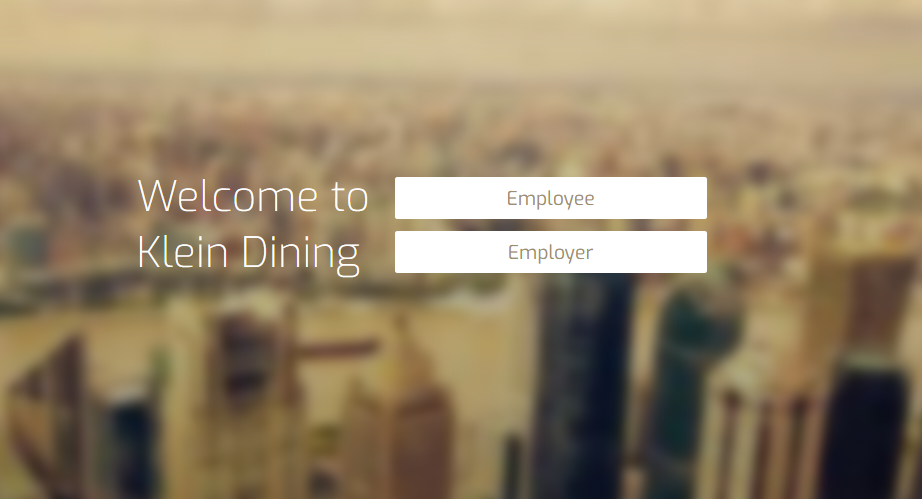
7) Upon clicking on the “submit” button, you will be redirected to the payroll list by period page where you can see all payroll items have been sent out to the respective employees. Click the “Home Page” button on the top left of the page to return to Employer navigation page.

Employees would receive an email notification regarding their payroll information as shown below.

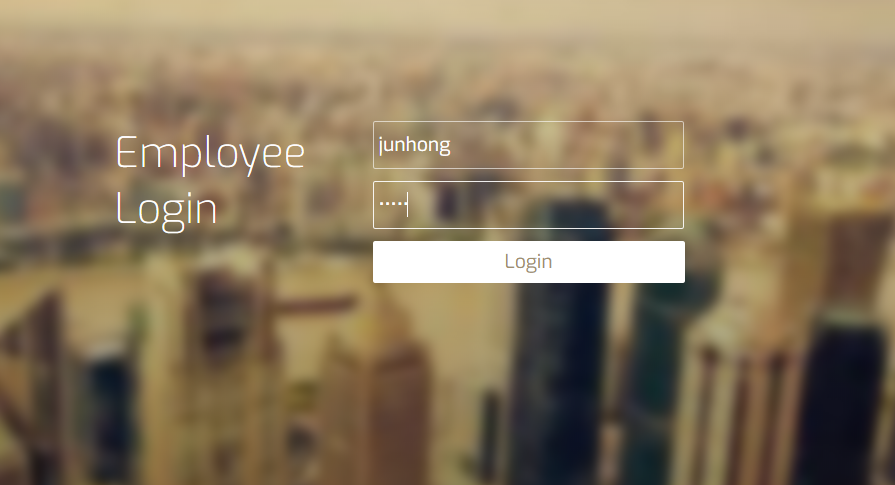


**

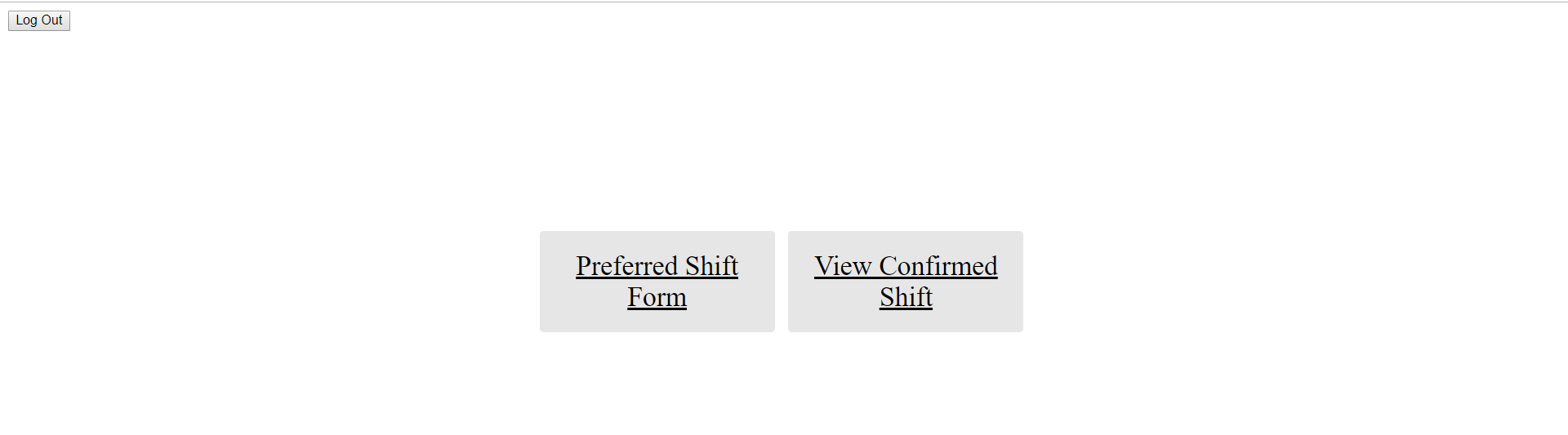
8) Click on the “Logout” button on the top left of the screen.



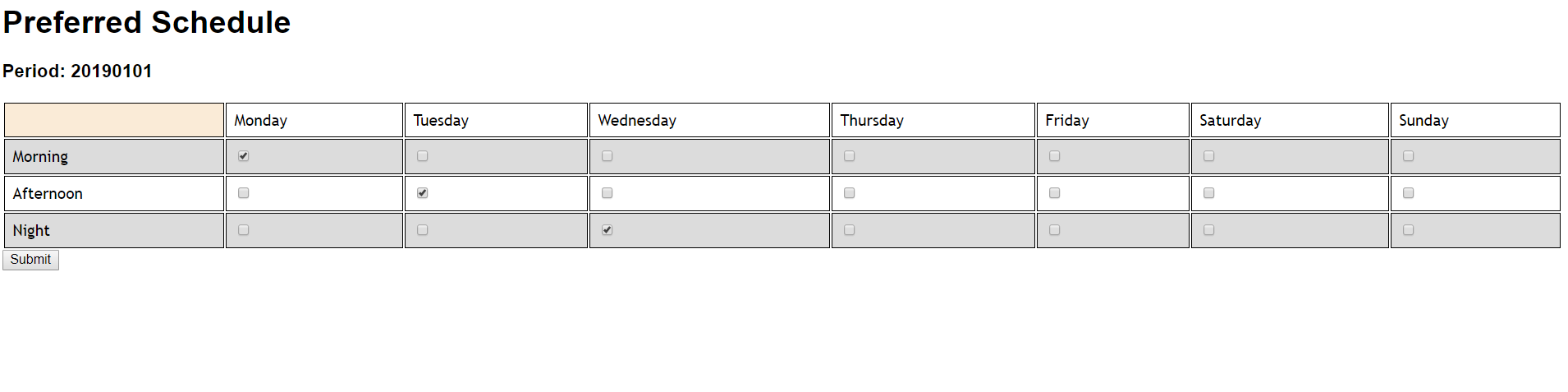
9) Click on the “Employee” button to access employee login user interface.



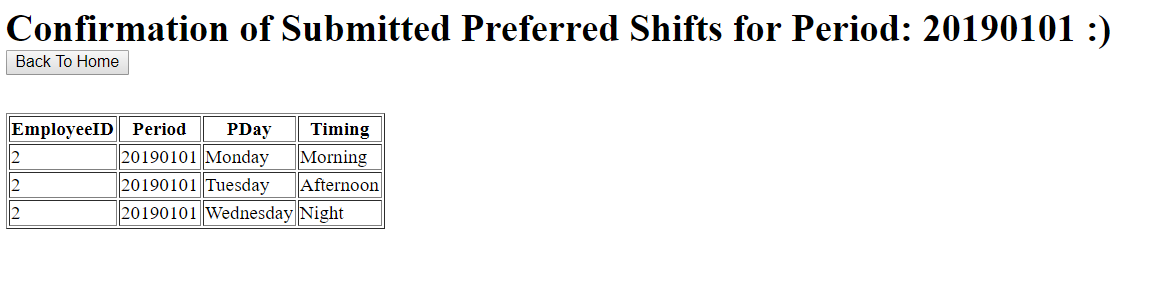
10) Login to Employee user interface with username: junhong and password: 12345.



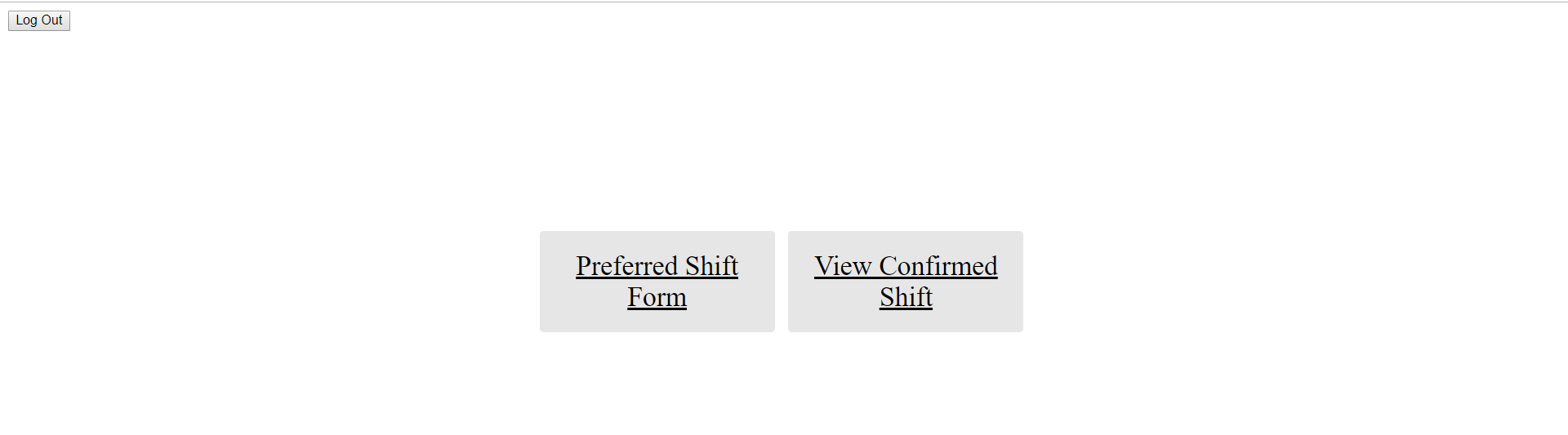
11) Click “Preferred Shift Form” button to access the preferred shift form.



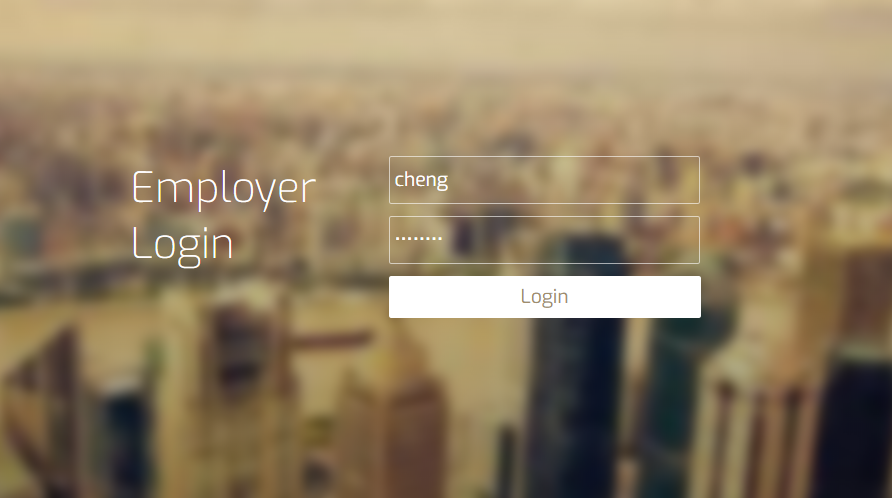
12) On this page, tick the checkboxes of your preferred shifts and click the “submit” button to submit preferred shifts to the employer.



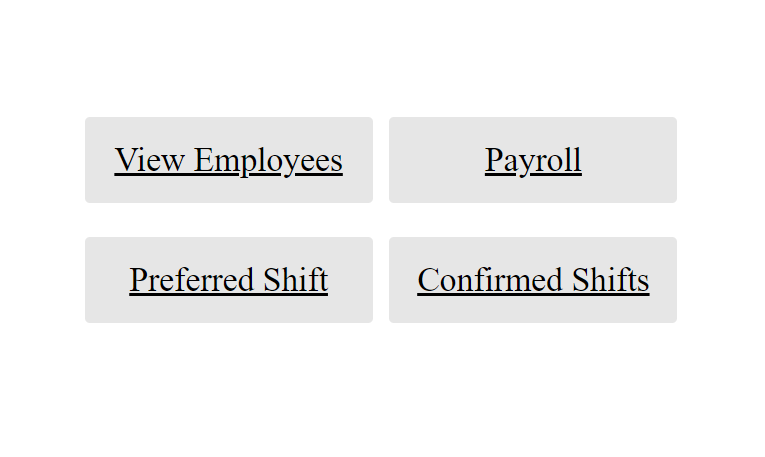
13) Upon submitting the form, you will be directed to this page where you can view the confirmation of the preferred shifts. Click “Back to Home” button to go back to employee navigation page.



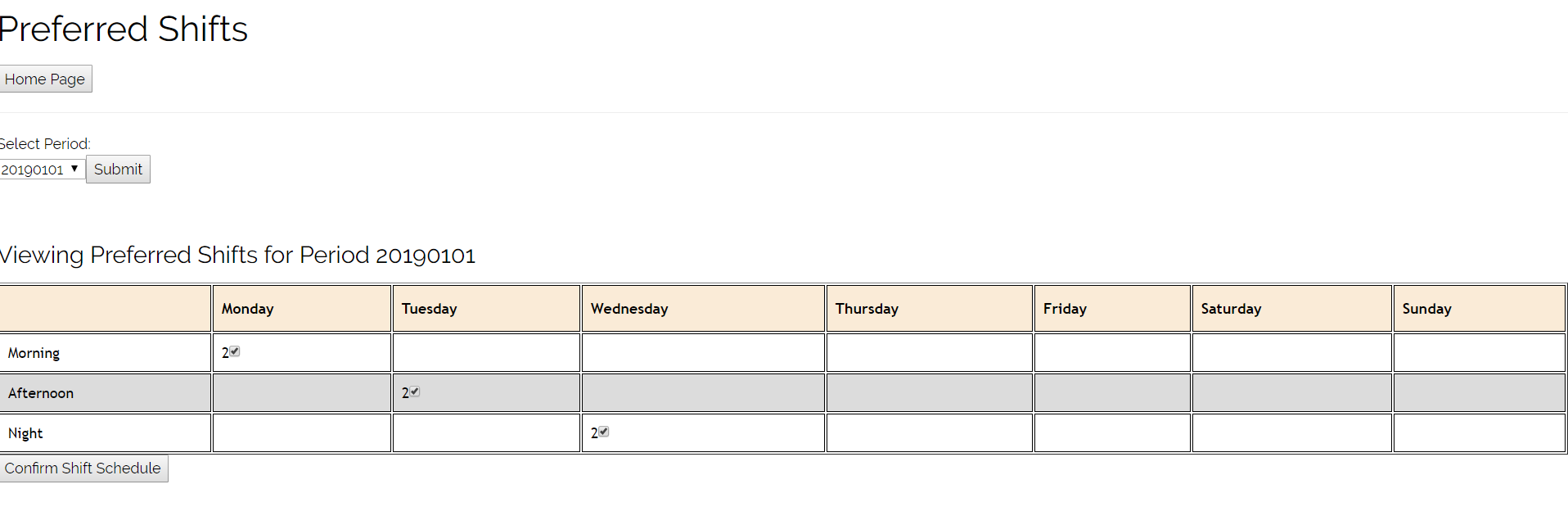
14) Click “Log Out” button on the top left of the screen.



15) Login to employer user interface once more with username: cheng and password: password

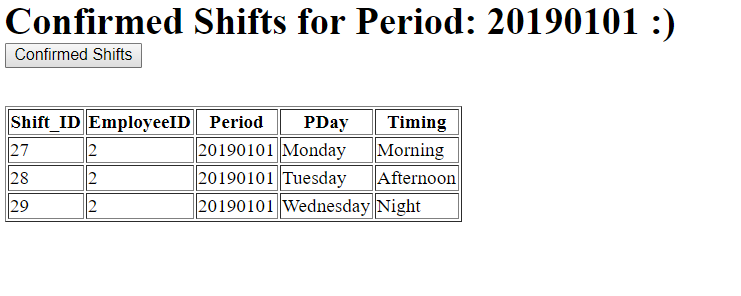


16) Click “Preferred Shift” to access the list of employees’ preferred shifts.



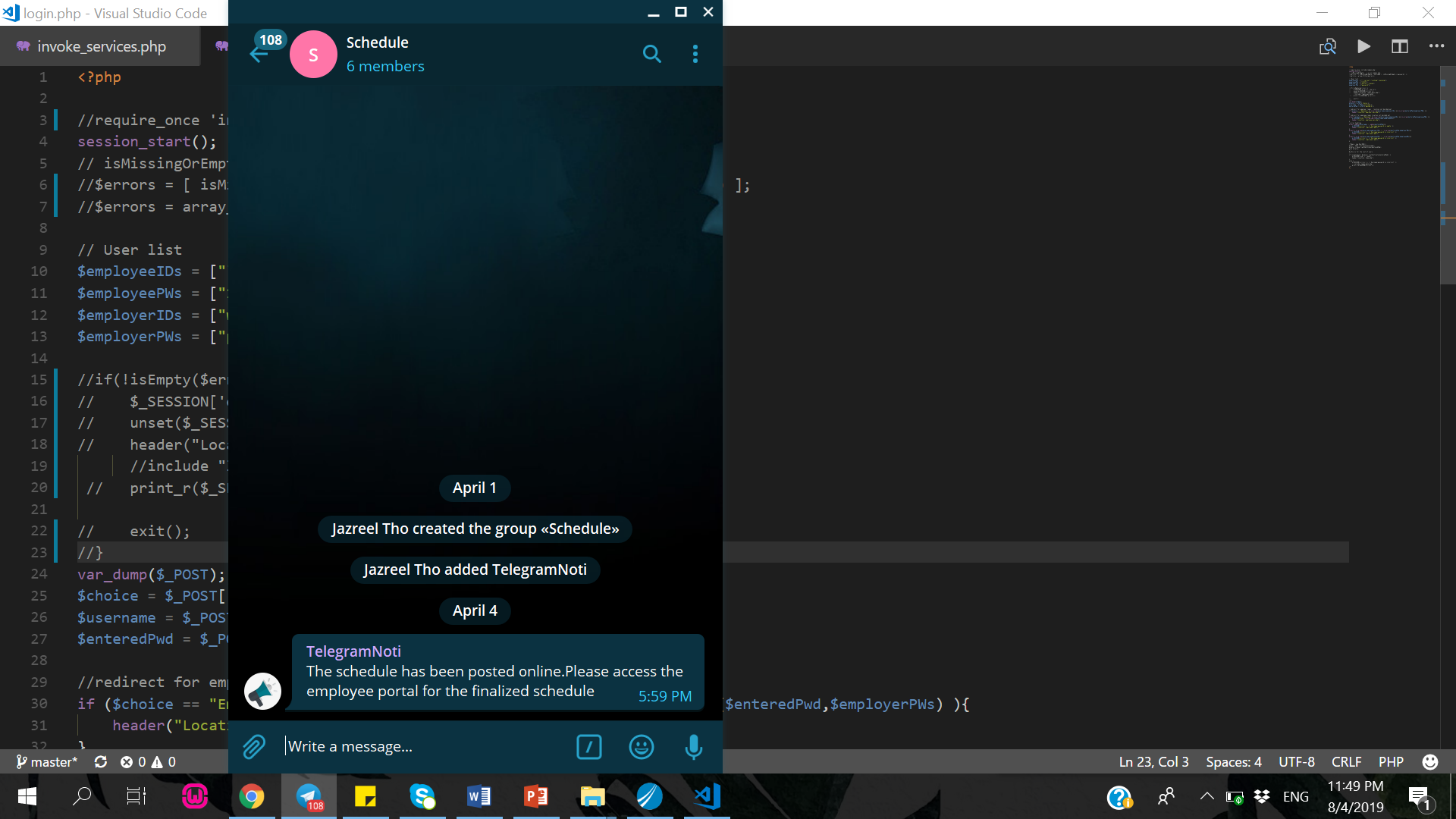
17) Select the period “20190101” and submit.

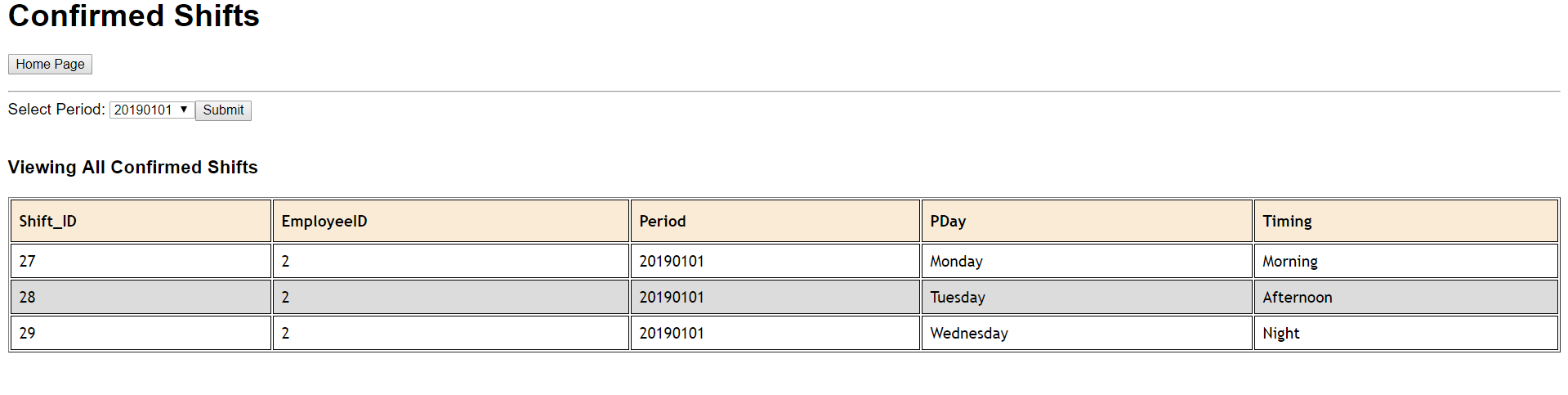
18) Tick the checkboxes for those shifts you wished to approve as shown in the picture above. Click on the “Confirm Shift Schedule” button once you are done approving the respective shifts.



19) Upon confirming the shift schedule, you will be able to see the shifts you approved as shown in the picture above. Click on “Confirmed Shifts” button.

Employees would receive a notification in their Telegram chat group regarding their confirmed schedule as shown below.

**



20) On this page, you will be able to see the confirmed shifts by period.

21) Click on “Home Page” and then “Log Out”. Done!