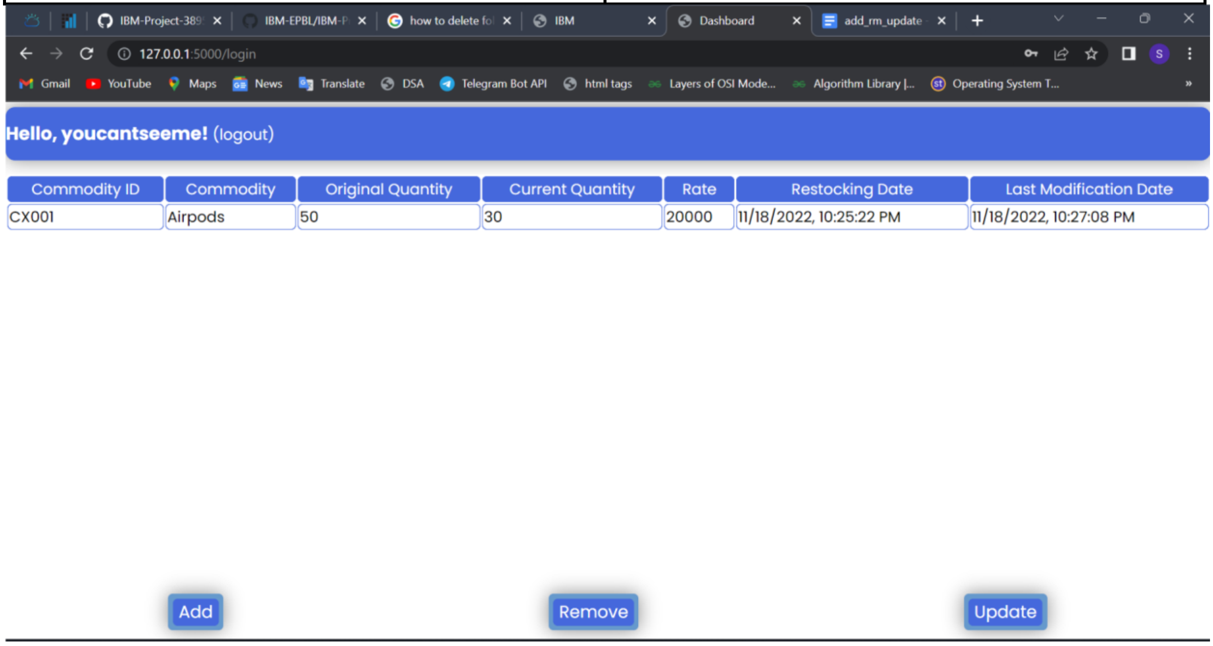
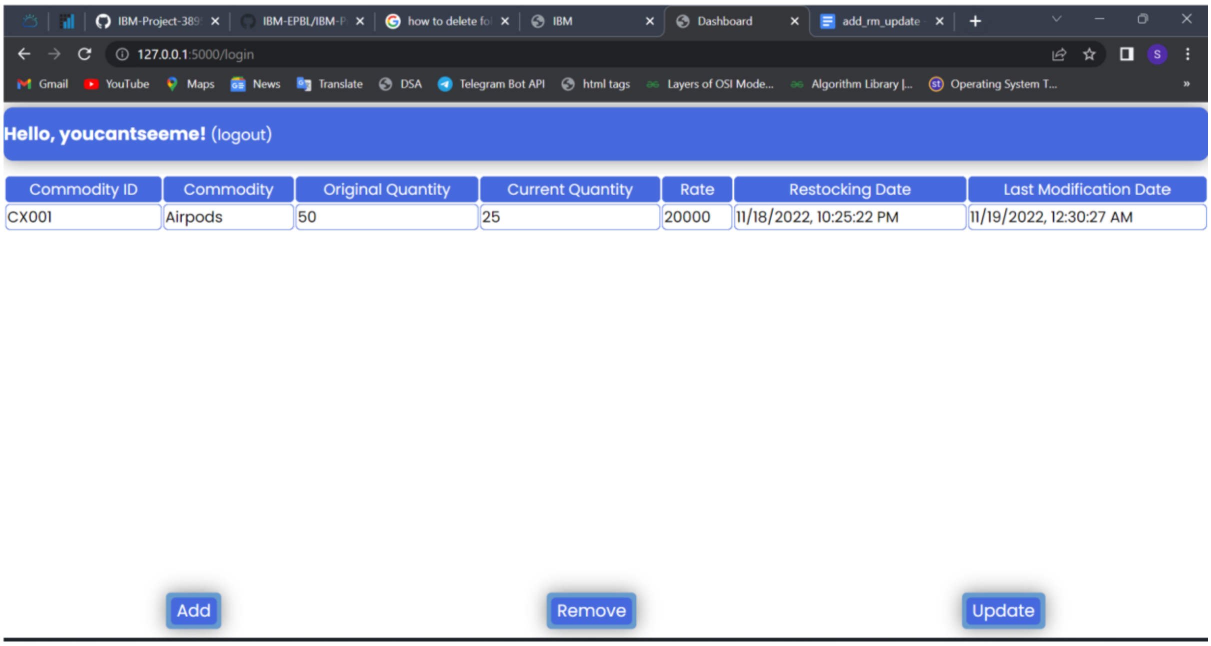
Login and Registration

Date	
Team ID	PNT2022TMID24193
Project Name	Inventory Management System for Retailers



Hello, youcantseeme! (logout)

Commodity ID	Commodity	Original Quantity	Current Quantity	Rate	Restocking Date	Last Modification Date
CX001	Airpods	50	30	20000	11/18/2022, 10:25:22 PM	11/18/2022, 10:27:08 PM



Hello, youcantseeme! (logout)

Commodity ID	Commodity	Original Quantity	Current Quantity	Rate	Restocking Date	Last Modification Date
CX001	Airpods	50	25	20000	11/18/2022, 10:25:22 PM	11/19/2022, 12:30:27 AM

Code

@app.route('/add-commodity', methods = ['POST']) def add_commodity(): # print(request.form)

```
    details = request.form.to_dict() tableContents =
    fetch_table_contents(session.get('username'))


    if tableContents[0][0] is None:
        tableContents = {field: [] for field in FIELDS}
    else: tableContents =
        loads(tableContents[0][0])

    if details.get('cid') in list(tableContents.values())[0]:
        return f"Item {details.get('cid')} already exists.", 400

    for key, value in tableContents.items():
        value.append(details.get(key))


    details_json = dumps(tableContents)
    print(details_json) print(details)


    try:
        add_details_to_db(session.get('username'), details_json)
    except Exception as exception:
        print(exception) return 'Something went
        wrong, try again', 403
    else:
        return 'Details Added.', 200
        # return redirect(url_for('login_check'), code = 307)

    return 'add req rcvd'


@app.route('/update-commodity', methods = ['POST'])
def update_commodity():
    details = request.form.to_dict() tableContents =
    fetch_table_contents(session.get('username'))

    cid = details.get('cid')
    qty = details.get('qty')
    date = details.get('date')
    if tableContents[0][0] is
    None:
        return 'The inventory is empty.', 403
    else: tableContents =
        loads(tableContents[0][0])

        try:
            tgtIndex = tableContents.get('cid').index(cid)
        except ValueError:
            return f'Item with code {cid} doesn\'t exist', 403
        else: if qty == '0' or qty
            == 0:
                try:
                    sendgrid_email('Restocking Reminder!', f'Item {cid} has run out, you should
probably restock it',
                                    'abc@def.com')
                except Exception as e:
                    print(f'Something went wrong -->
                    {e}')

            if int(qty) > int(tableContents.get('cqty')[tgtIndex]):
                tableContents.get('oqty')[tgtIndex] = qty
                tableContents.get('date')[tgtIndex] = date
            else: tableContents.get('date_mdf')[tgtIndex] =
                date


            tableContents.get('cqty')[tgtIndex] = qty


            tableContents = dumps(tableContents)

            try:
                add_details_to_db(session.get('username'), tableContents)
            except Exception as exception:
                print(exception) return 'Something went
                wrong, try again', 403
            else:
                return 'Details Updated.', 200
            return 'Updated', 200

    return 'Huh?', 403



@app.route('/remove-commodity', methods = ['POST'])
def remove_commodity():
    details = request.form.to_dict()
    tableContents = fetch_table_contents(session.get('username'))

    cid = details.get('cid')

    if tableContents[0][0] is None:
        return 'The inventory is empty.', 403
```

```
else: tableContents =
    loads(tableContents[0][0])

    try:
        tgtIndex = tableContents.get('cid').index(cid)
    except ValueError:
        return f'Item with code {cid} doesn\'t exist', 403
    else:
        for key, value in tableContents.items():
            value.pop(tgtIndex)
    tableContents = dumps(tableContents)

    try:
        add_details_to_db(session.get('username'), tableContents)
    except Exception as exception:
        print(exception)
        return 'Something went wrong, try again', 403
    else: return f'Item {cid} removed.',
        200
```

Some other codes like handling verification in the front end etc. have been skipped for the sake of clarity.