**Helpful Instructions for Creating a calendar view Dashboard from a database.**

**Input database:**

This database is a list of students who have reserved slots (slot\_id). These reservations are done by the users (user\_id). Here our goal is to make a dashboard from this database. The range of slots is from 2 to 16 (15 fixed slots for each data). Hence, any user can book a slot (among 15 slots) for any date. Each Slot and Date is unique. A single slot on a particular date cannot be reserved by any student. The User ID is used when a user logged into the application and reserved a slot. Without user ID no slot can be reserved.

A table with text and numbers

Description automatically generated

**Output Dashboard:**

A screen shot of a computer

Description automatically generated

The Pseudocode for extracting data from database in the views:

* Retrieves all lab rooms from the database (query: labroom)
* Obtains the current date (var: today)
* Create a list of dates for the next seven days starting from today (list: dates)
* Creates a list of lab rooms for iteration (query: labroom > list: labroom\_list)
* Create a list of dictionaries (list: urls[{view\_url},{update\_url}{reserve\_url}])
* Generate the lab room detail URL for a specific lab room and date for view url for each slot no matter it is reserved/available. (dict: view\_url)
* Generates update and reserve URLs based on whether a student has reserved a slot for the lab room and date. (dict: {update\_url} or dict: {reserve\_url})
* A loop iterates over each lab room and date to populate the list.
* The table\_data list stores dictionaries with lab room information and their corresponding URLs for each date (list: table\_data[list:urls])
* The dictionary (dict:context) contains the data to be passed to the template (list:dates, list:labroom\_list, list: table\_data[list:urls])
* Finally, the template is rendered with the provided context.

The view function is as below:

|  |
| --- |
| def labroom\_list(request):      labrooms = LabRoom.objects.all()      today = datetime.now().date()      dates = []      labroom\_list = [labroom for labroom in labrooms]      table\_data = []      for i in range(7):          dates.append(today)          today = datetime.now().date() + timedelta(days=i+1)      view\_urls\_datewise = []      for labroom in labrooms:          today = datetime.now().date()          temp = []          for i in range(7):              vur\_url = {}              vur\_url["view"] = reverse('labroom-detail', args=[labroom.pk, str(today)])              student = Student.objects.filter(slot\_id = labroom.id, date=today).values().all()              if student.exists():                  vur\_url["update"] = (reverse('update\_slot', args=[labroom.pk, str(today)]))              else:                  vur\_url["reserve"] = (reverse('reserve\_slot', args=[labroom.pk, str(today)]))              temp.append(vur\_url)                today = datetime.now().date() + timedelta(days=i+1)          view\_urls\_datewise.append(temp)          table\_data.append({'labroom':labroom,'urls':temp})      context = {'views\_url': view\_urls\_datewise,'dates':dates,'labrooms':labroom\_list,'data':table\_data}      return render(request, 'alras\_application/labroom\_list.html', context) |

The pseudocode for displaying the dashboard in html:

* Iterate through the dates to generate the table headers.
* Iterate through the row to generate the data rows.
* The first column in the data row prints the room slots.
* The later columns in the data row print the URLs.
* Use a if statement to choose the cell color depending on if the url is for update or reserve.

The html template is as below:

|  |
| --- |
| {% block content %}  <div class="container justify-content-center">  <h1 style="color:tomato;">Cyber Security Lab Room Availability Status</h1>  </div>  <div class="table-responsive">  {% if data %}  <table class="table table-bordered border-primary">      <thead class="table-dark">          <tr>              <th scope="col">Room/Slot</th>              {% for date in dates %}                  <th scope="col">{{date|date:"d M Y" }}</th>              {% endfor %}          </tr>      </thead>      <tbody>          {% for x in data %}          <tr>              <th class="table-dark-bg-subtle" scope="row">{{x.labroom}}</th>              {% for url in x.urls %}                  {% if url.reserve %}                  <td class="table-danger"><a class="btn btn-info btn-sm" href="{{ url.view }}" role="button">View</a>                     <a class="btn btn-secondary btn-sm" href="{{ url.reserve }}" role="button">Reserve</a></td>                  {% else %}                      <td class="table-warning"><a class="btn btn-info btn-sm" href="{{ url.view }}" role="button">View</a>                      <a class="btn btn-success btn-sm" href="{{ url.update }}" role="button">Update</a></td>                  {% endif %}          {% endfor %}          </tr>          {% endfor %}    </tbody>  </table>  {% else %}  <p>There are no lab rooms configured.</p>  </div>  {% endif %}  </div>  {% endblock %} |