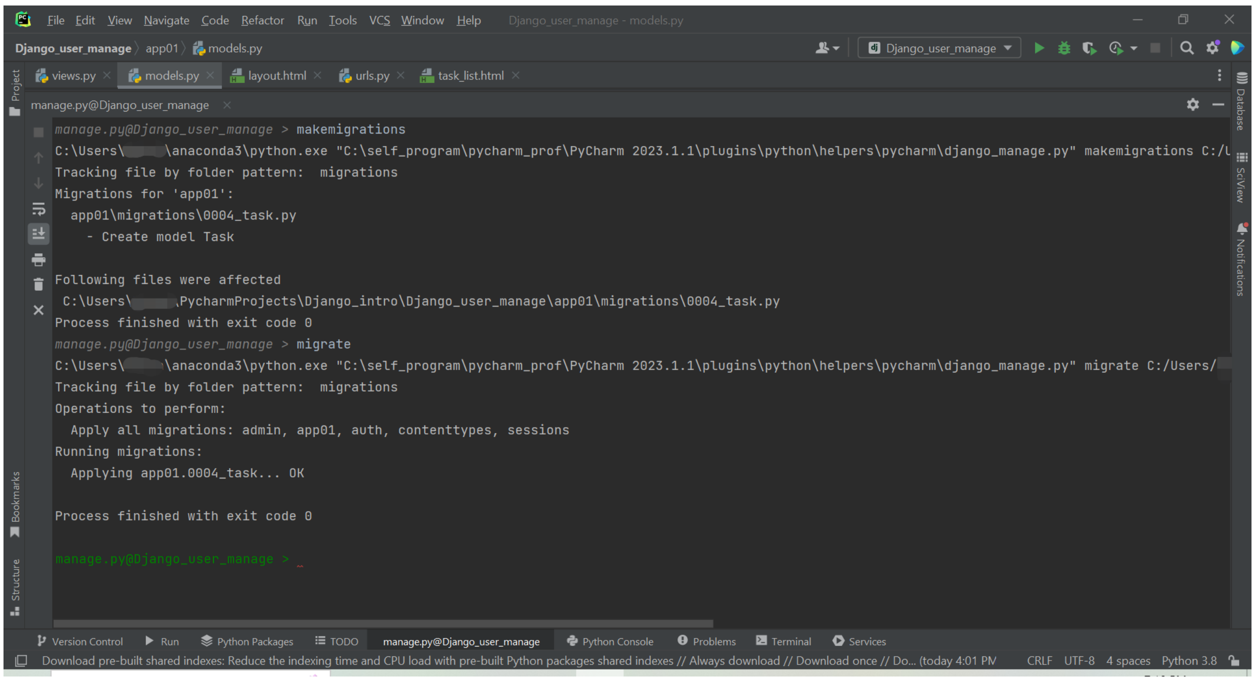
ajax example

1. create table

```
class Task(models.Model):
    title = models.CharField(verbose_name="TaskTitle", max_length=64)
    detail = models.TextField(verbose_name="Detail")
    level_choices = {
        (1, 'serious'),
        (2, 'important'),
        (3, 'temporary'),
    }
    user = models.ForeignKey(verbose_name='Principal', to="Admin", on_delete=models.CASCADE)
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help Django_user_manage - models.py
Django_user_manage 〉 app01 〉 models.py
Django_user_manage
views.py
models.py
layout.html
urls.py
task_list.html
manage.py@Django_user_manage
Project
Database
SciView
Notifications
Bookmarks
Structure
Version Control
Run
Python Packages
TODO
manage.py@Django_user_manage
Python Console
Problems
Terminal
Services
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Do... (today 4:01 PM
CRLF
UTF-8
4 spaces
Python 3.8

```
manage.py@Django_user_manage > makemigrations
C:\Users\[redacted]\anaconda3\python.exe "C:\self_program\pycharm_prof\PyCharm 2023.1.1\plugins\python\helpers\pycharm\django_manage.py" makemigrations C:/U
Tracking file by folder pattern:  migrations
Migrations for 'app01':
  app01\migrations\0004_task.py
    - Create model Task

Following files were affected
 C:\Users\[redacted]\PycharmProjects\Django_intro\Django_user_manage\app01\migrations\0004_task.py
Process finished with exit code 0
manage.py@Django_user_manage > migrate
C:\Users\[redacted]\anaconda3\python.exe "C:\self_program\pycharm_prof\PyCharm 2023.1.1\plugins\python\helpers\pycharm\django_manage.py" migrate C:/Users/[redacted]
Tracking file by folder pattern:  migrations
Operations to perform:
  Apply all migrations: admin, app01, auth, contenttypes, sessions
Running migrations:
  Applying app01.0004_task... OK

Process finished with exit code 0

manage.py@Django_user_manage >
```

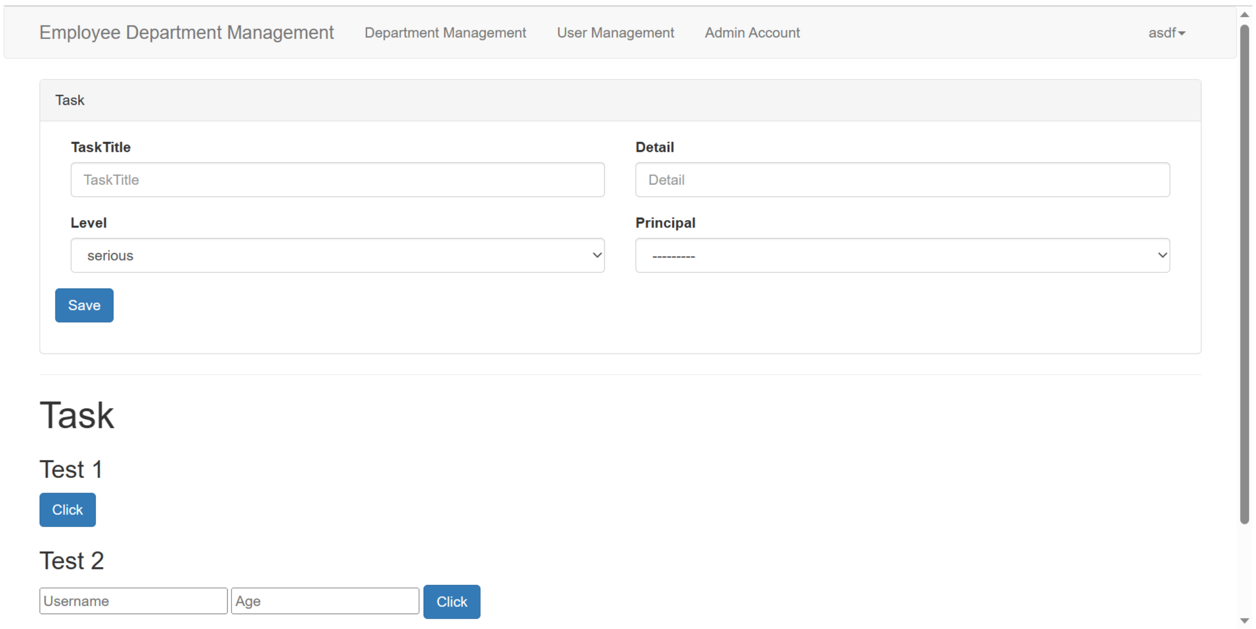
2. create model form for Task

```
class Admin(models.Model):
    username = models.CharField(verbose_name='Admin_name', max_length=32)
    password = models.CharField(verbose_name='Password', max_length=64)

    def __str__(self):
        return self.username
```

```
class TaskModelForm(BootStrapModelForm):
    class Meta:
        model = models.Task
        fields = "__all__"
        widgets = {
            # "detail": forms.Textarea
            "detail": forms.TextInput,
        }
```

```
{% extends 'layout.html' %}

{% block content %}
    <div class="container">

        <div class="panel panel-default">
```

```
            <div class="panel-heading">Task</div>
            <div class="panel-body">
                <form method="post" novalidate>
                    <div class="clearfix">
                        {% for field in form %}
                            <div class="col-xs-6">
                                <div class="form-group">
                                    <label>{{ field.label }}</label>
                                    {{ field }}
                                </div>
                            </div>
                        {% endfor %}
                    </div>
                    <div class="form-group">
                        <button type="submit" class="btn btn-primary">Save</button>
                    </div>
                </form>
            </div>
        </div>


        <hr/>
        <h1>Task</h1>
        <h3>Test 1</h3>
        <input id="btn1" type="button" class="btn btn-primary" value="Click">

        <h3>Test 2</h3>
        <input id="textUser1" type="text" placeholder="Username">
        <input id="textAge1" type="text" placeholder="Age">
        <input id="btn2" type="button" class="btn btn-primary" value="Click">

        <h3>Test 3</h3>
        <form id="form1">
            <input name="User2" type="text" placeholder="Username">
            <input name="Age2" type="text" placeholder="Age">
            <input name="Email2" type="text" placeholder="Email">
            <input name="Mobile2" type="text" placeholder="Mobile">
            <input name="More2" type="text" placeholder="More">

        </form>
        <input id="btn3" type="button" class="btn btn-primary" value="Click">

    </div>
{% endblock %}

{% block js %}

    <script type="text/javascript">
        $(function () {
            bindBtn1Event();

            bindBtn2Event();

            bindBtn3Event();

        })

        function bindBtn1Event() {
            $("#btn1").click(function () {
                $.ajax({
                    url: '/task/ajax/',
                    type: "post",
                    data: {
                        n1: 123,
                        n2: 456
                    },
                    dataType: "JSON",
                    success: function (res) {
                        console.log(res);
                        console.log(res.status);
                        console.log(res.data);
                    }
                })
            })
        }

        function bindBtn2Event() {
            $("#btn2").click(function () {
                $.ajax({
                    url: '/task/ajax/',
                    type: "post",
                    data: {
                        status: 'aaaaaaaaa',
                        name: $("#textUser1").val(),
                        age: $("#textAge1").val()
                    },
                    dataType: "JSON",
                    success: function (res) {
                        console.log(res);
                        console.log(res.status);
                        console.log(res.data);
                        console.log(res.name);
                    }
                })
            })
        }

        function bindBtn3Event() {
            $("#btn3").click(function () {
                $.ajax({
                    url: '/task/ajax/',
                    type: "post",
                    data: $("#form1").serialize(),
                    dataType: "JSON",
                    success: function (res) {
                        console.log(res);
                        console.log(res.status);
                        console.log(res.data);
                        console.log(res.name);
                    }
                })
            })
        }

        function clickMe() {
            $.ajax({
                url: '/task/ajax/',
                type: "post",
                data: {
                    n1: 123,
                    n2: 456
                }, success: function (res) {
                    console.log(res);
                }
            })
        }
    </script>
{% endblock %}
```



3. ajax

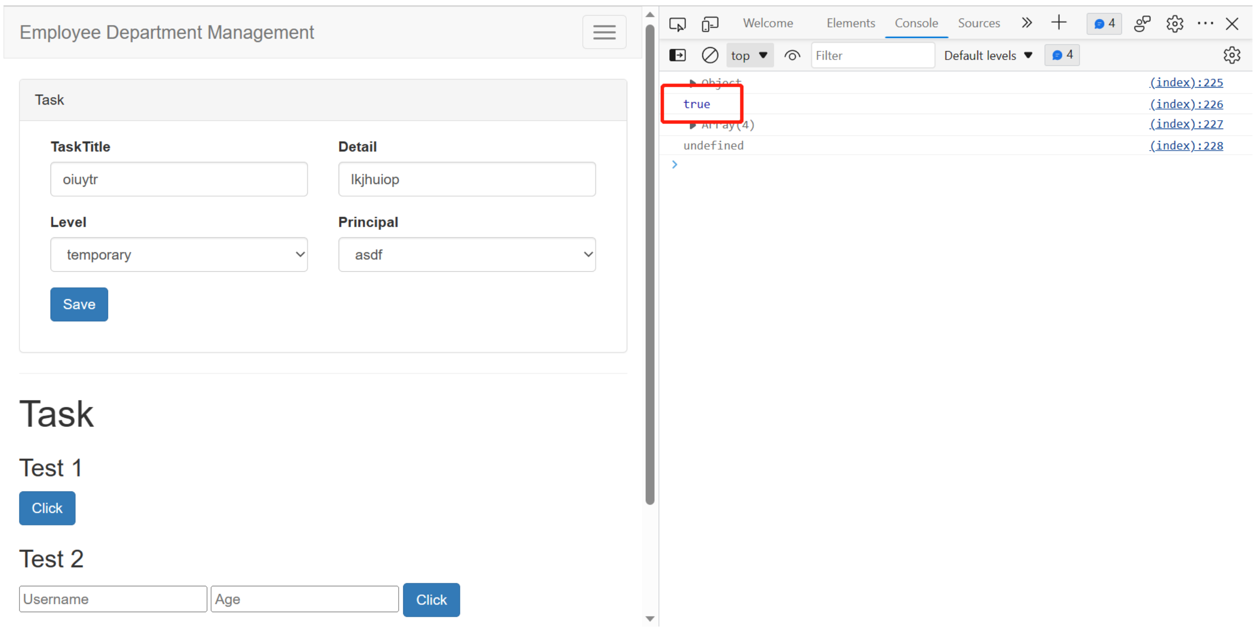
add url

```
path('task/add/', views.task_add),
```

add function

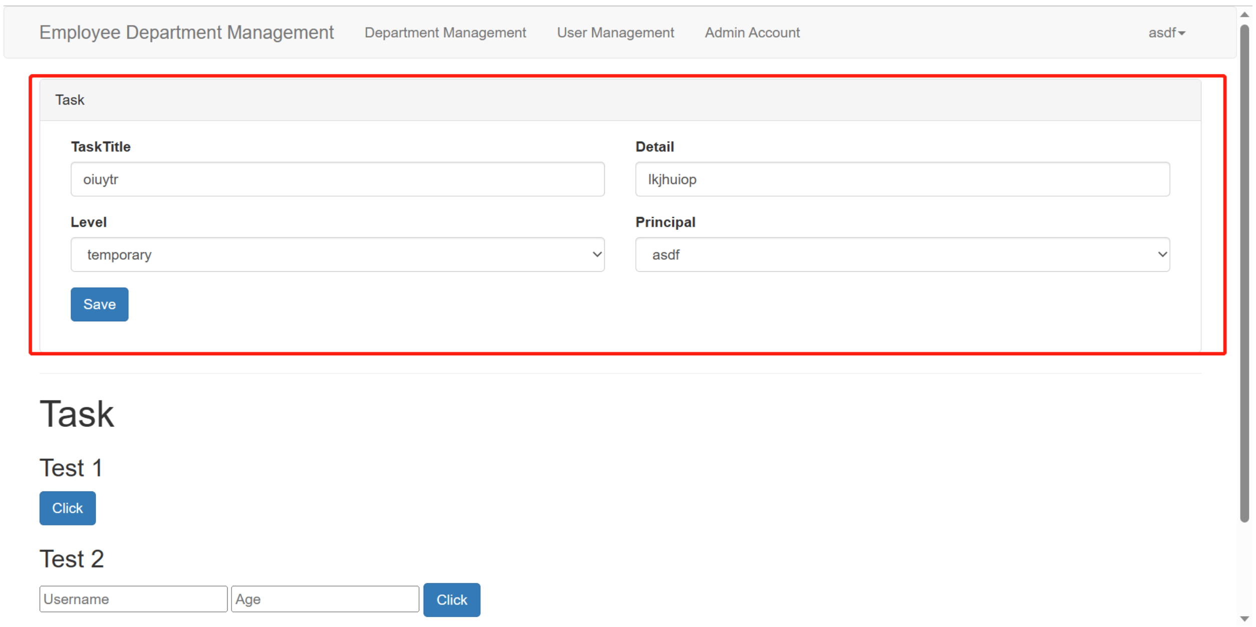
```
function bindBtnAddEvent() {
            $("#btnAdd").click(function () {
                $.ajax({
                    url: '/task/add/',
                    type: "post",
                    data: $("#formAdd").serialize(),
                    dataType: "JSON",
                    success: function (res) {
                        console.log(res);
                        console.log(res.status);
                        console.log(res.data);
                        console.log(res.name);
                    }
                })
            })
        }
```

```
@csrf_exempt
def task_add(request):

    print(request.POST)

    data_dict = {"status": True}

    return HttpResponse(json.dumps(data_dict))
```



TaskTitle: oiuytr
Detail: lkjhuiop
Level: temporary
Principal: asdf
Username
Age

```
▶ Object                (index):225
true                    (index):226
▶ Array(4)              (index):227
undefined               (index):228
>
```



Task

TaskTitle: oiuytr

Detail: lkjhuiop

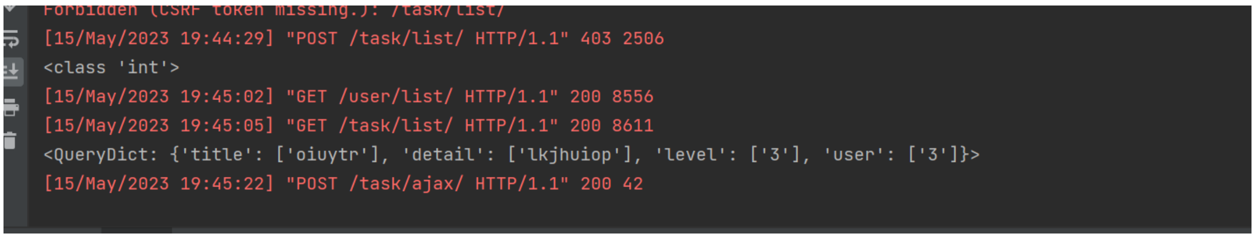
Level: temporary

Principal: asdf

Task

Test 1

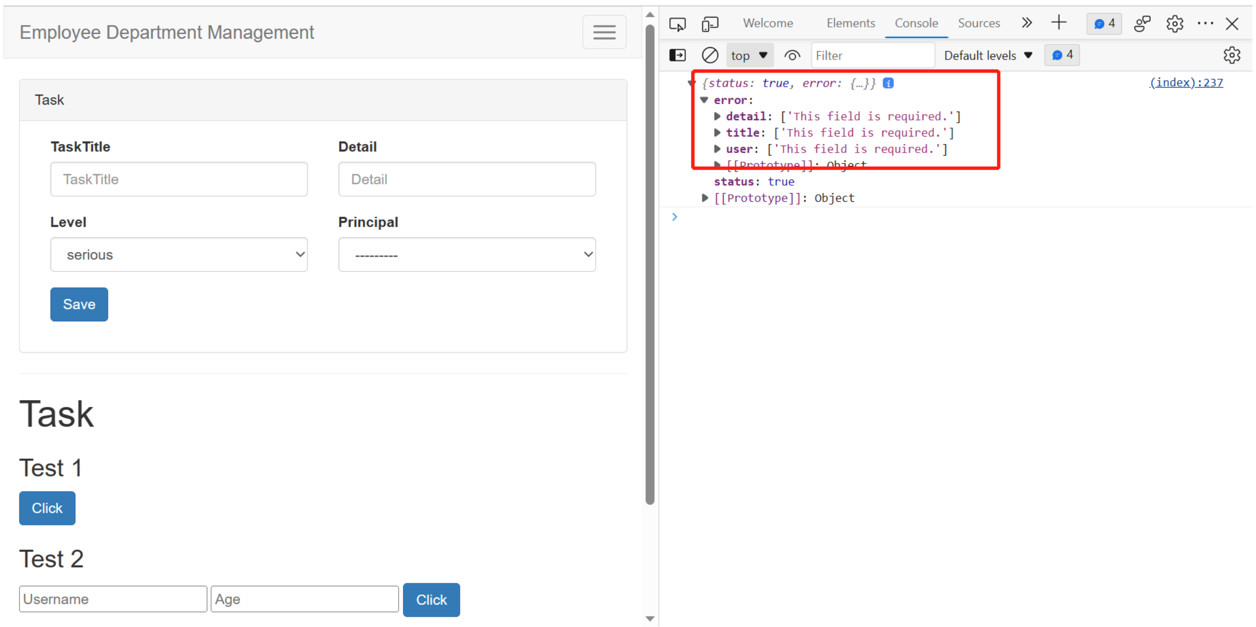
Test 2



```
Forbidden (CSRF token missing.): /task/list/
[15/May/2023 19:44:29] "POST /task/list/ HTTP/1.1" 403 2506
<class 'int'>
[15/May/2023 19:45:02] "GET /user/list/ HTTP/1.1" 200 8556
[15/May/2023 19:45:05] "GET /task/list/ HTTP/1.1" 200 8611
<QueryDict: {'title': ['oiuytr'], 'detail': ['lkjhuiop'], 'level': ['3'], 'user': ['3']}>
[15/May/2023 19:45:22] "POST /task/ajax/ HTTP/1.1" 200 42
```

error message

```
@csrf_exempt
def task_add(request):

    print(request.POST)
    # method 1
    # Use ModelForm

    form = TaskModelForm(data=request.POST)

    if form.is_valid():
        form.save()
        data_dict = {"status": True}
        return HttpResponse(json.dumps(data_dict))


    data_dict = {"status": True, 'error': form.errors}

    return HttpResponse(json.dumps(data_dict, ensure_ascii=False))
```



Task

TaskTitle: TaskTitle

Detail: Detail

Level: serious

Principal: ---------

Task

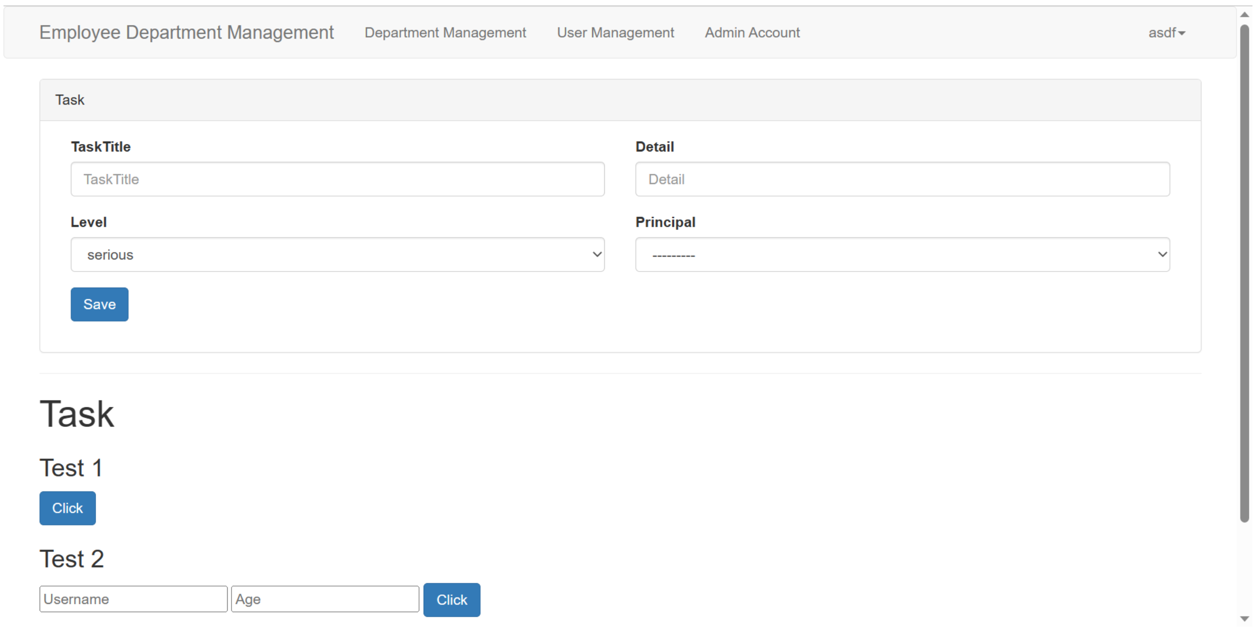
Test 1

Test 2

```
▼ {status: true, error: {…}}                    (index):237
  ▼ error:
    ▶ detail: ['This field is required.']
    ▶ title: ['This field is required.']
    ▶ user: ['This field is required.']
    ▶ [[Prototype]]: Object
    status: true
  ▶ [[Prototype]]: Object
>
```

show error message

need to set position relative & absolute, to avoid:



TaskTitle: TaskTitle

Detail: Detail

Level: serious

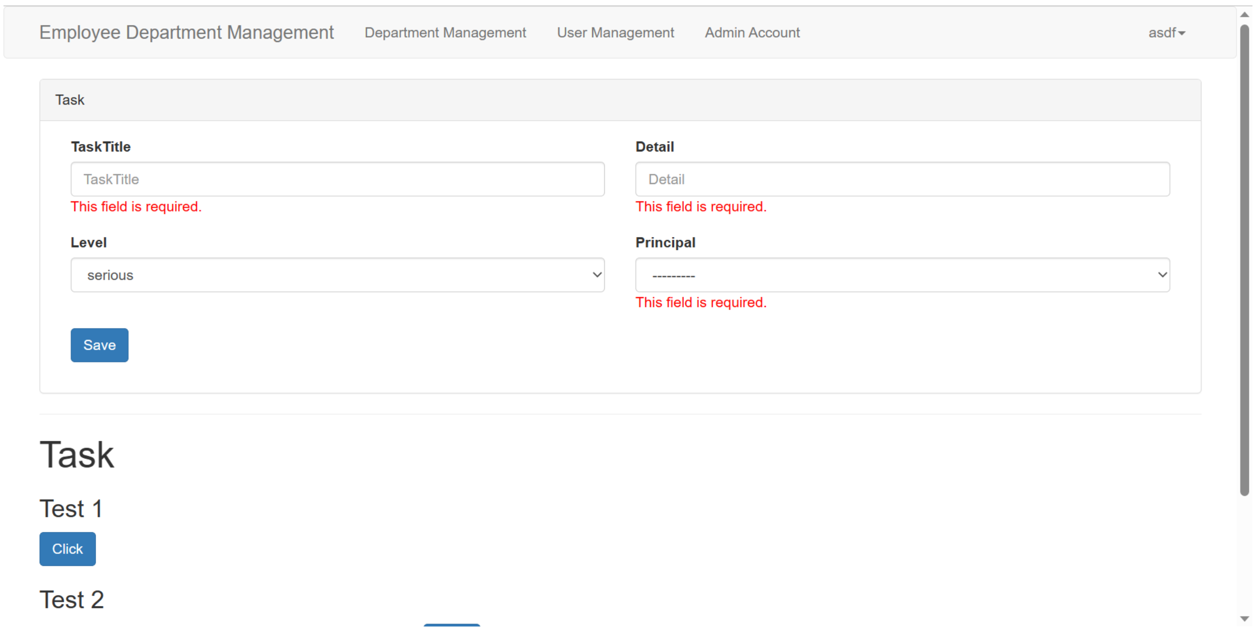
Principal: ---------

Task

Test 1

Test 2

Username Age



TaskTitle: TaskTitle

This field is required.

Detail: Detail

This field is required.

Level: serious

Principal: ---------

This field is required.

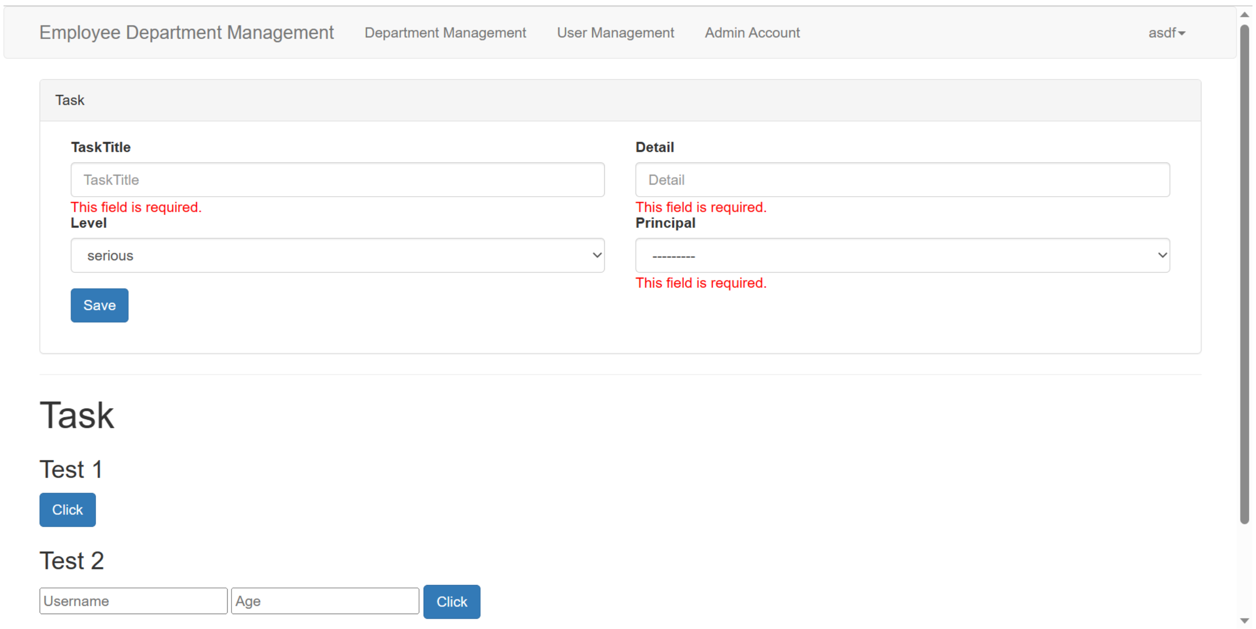
Task

Test 1

Test 2

panel gets larger (size change)

```
{% extends 'layout.html' %}

{% block content %}
    <div class="container">

        <div class="panel panel-default">
            <div class="panel-heading">Task</div>
            <div class="panel-body">
                <form id="formAdd">
                    <div class="clearfix">
                        {% for field in form %}
                            <div class="col-xs-6">
                                <div class="form-group" style="position: relative">
                                    <label>{{ field.label }}</label>
                                    {{ field }}
                                    <span style="color: red; position: absolute"></span>
                                </div>
                            </div>
                        {% endfor %}
                    </div>
                    <div class="col-xs-12">
                        <div class="form-group">
                            <button id="btnAdd" type="button" class="btn btn-primary">Save</button>
                        </div>
                    </div>
                </form>
            </div>
        </div>




    </div>
{% endblock %}

{% block js %}

    <script type="text/javascript">
        $(function () {

            bindBtnAddEvent();

        })



        function bindBtnAddEvent() {
            $("#btnAdd").click(function () {
                $.ajax({
                    url: '/task/add/',
                    type: "post",
                    data: $("#formAdd").serialize(),
                    dataType: "JSON",
                    success: function (res) {
                        if(res.status){
                            alert('Add Successful! ^-^')
                        }else{
                            $.each(res.error, function(name, data) {
                                    console.log(name, data);
                                    $("#id_"+name).next().text(data[0])
                                })
                        }
                    }
                })
            })
        }


    </script>
{% endblock %}
```



TaskTitle

TaskTitle

This field is required.

Detail

Detail

This field is required.

Level

serious

Principal

This field is required.

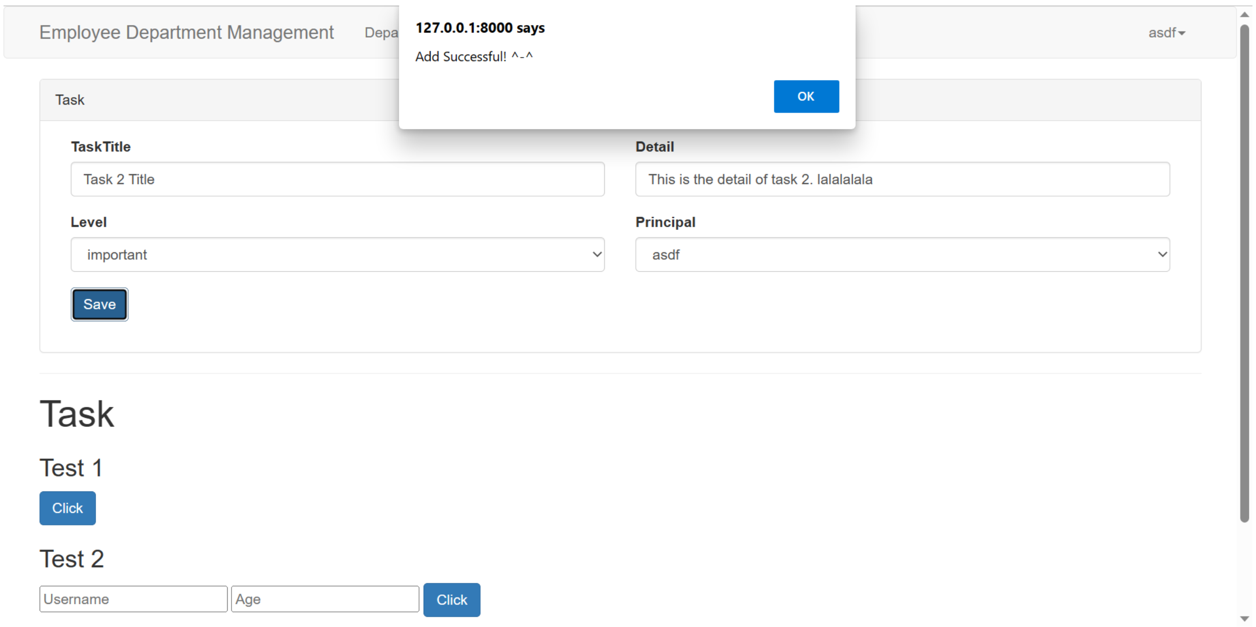
Task

Test 1

Test 2

Username

Age



127.0.0.1:8000 says

Add Successful! ^-^

Task

TaskTitle: Task 2 Title

Detail: This is the detail of task 2. lalalalala

Level: important

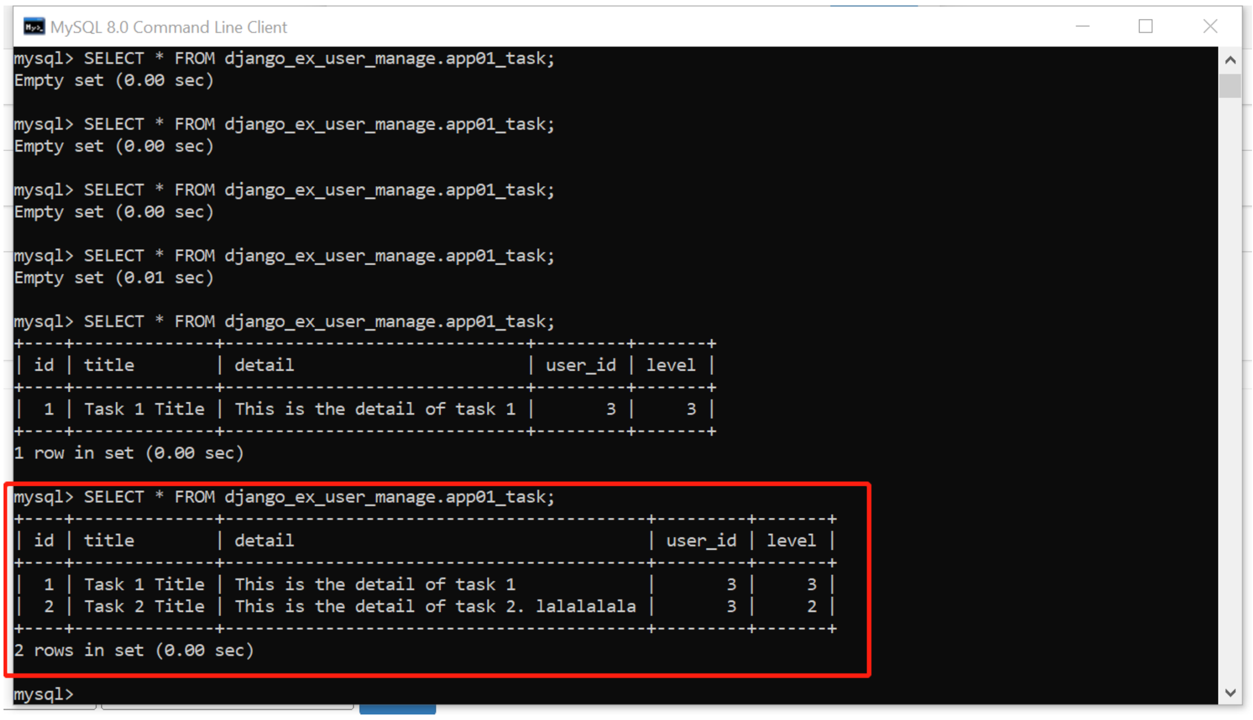
Principal: asdf

Task

Test 1

Test 2

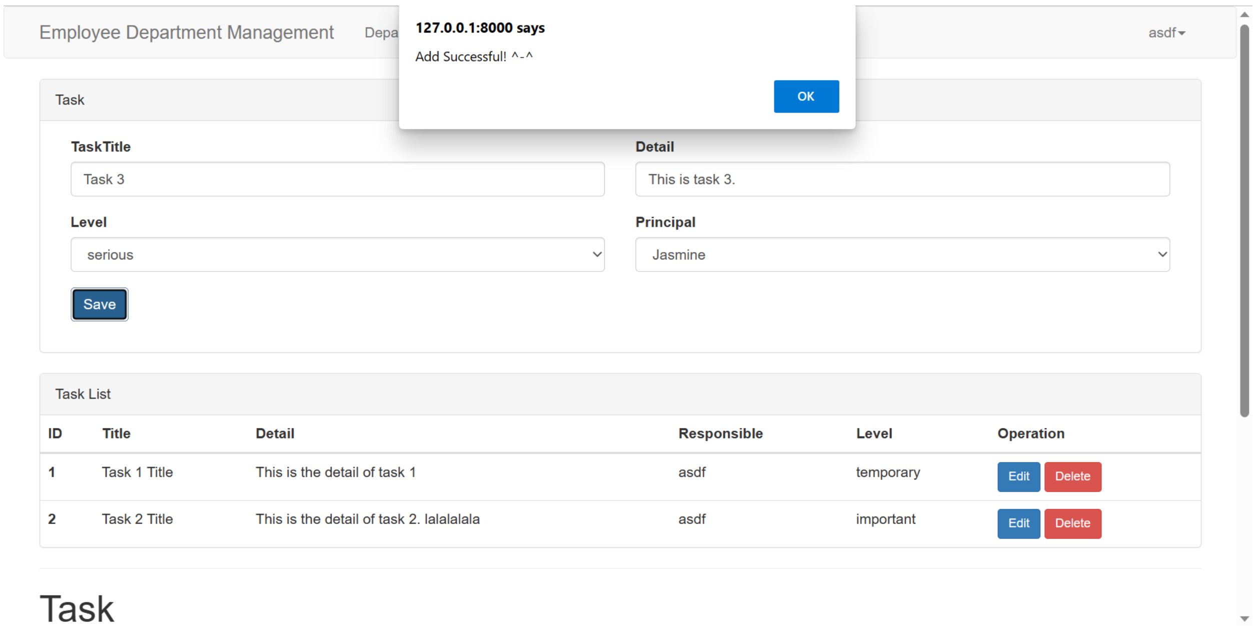
Username | Age



```
mysql> SELECT * FROM django_ex_user_manage.app01_task;
Empty set (0.00 sec)

mysql> SELECT * FROM django_ex_user_manage.app01_task;
Empty set (0.00 sec)

mysql> SELECT * FROM django_ex_user_manage.app01_task;
Empty set (0.00 sec)

mysql> SELECT * FROM django_ex_user_manage.app01_task;
Empty set (0.01 sec)

mysql> SELECT * FROM django_ex_user_manage.app01_task;
+----+--------------+------------------------------+---------+-------+
| id | title        | detail                       | user_id | level |
+----+--------------+------------------------------+---------+-------+
|  1 | Task 1 Title | This is the detail of task 1 |       3 |     3 |
+----+--------------+------------------------------+---------+-------+
1 row in set (0.00 sec)

mysql> SELECT * FROM django_ex_user_manage.app01_task;
+----+--------------+-----------------------------------------+---------+-------+
| id | title        | detail                                  | user_id | level |
+----+--------------+-----------------------------------------+---------+-------+
|  1 | Task 1 Title | This is the detail of task 1            |       3 |     3 |
|  2 | Task 2 Title | This is the detail of task 2. lalalalala |       3 |     2 |
+----+--------------+-----------------------------------------+---------+-------+
2 rows in set (0.00 sec)

mysql>
```

add task list

When add, task list only shown new data after refresh.



127.0.0.1:8000 says

Add Successful! ^-^

Task

TaskTitle: Task 3

Detail: This is task 3.

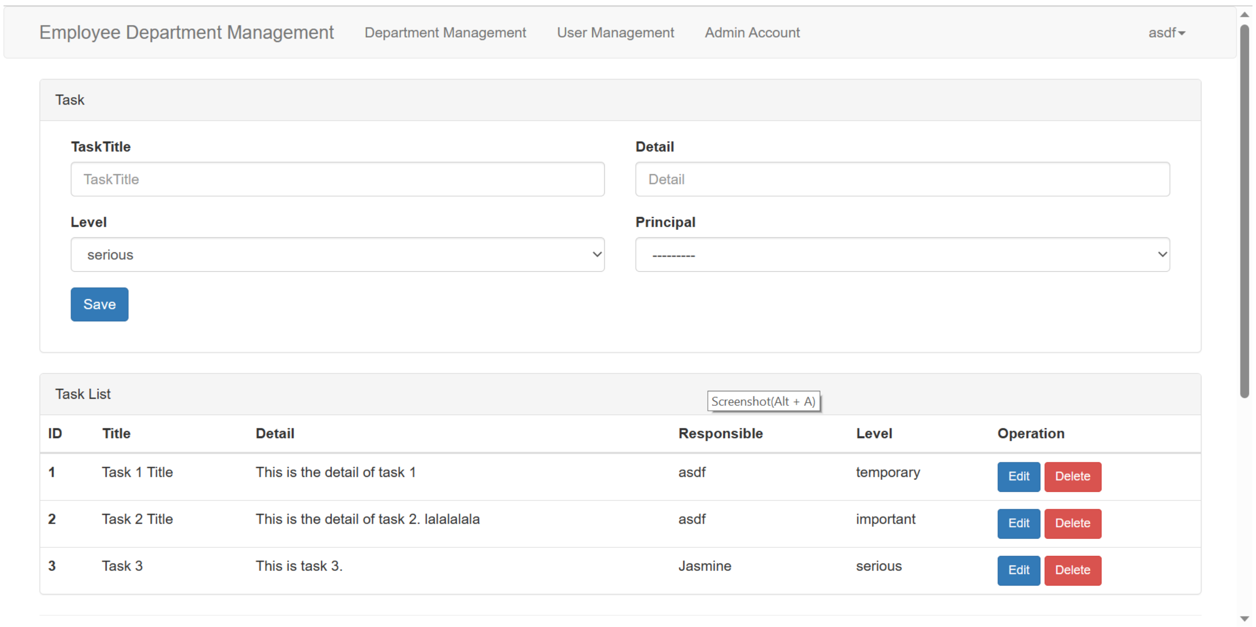
Level: serious

Principal: Jasmine

Task List

ID	Title	Detail	Responsible	Level	Operation
1	Task 1 Title	This is the detail of task 1	asdf	temporary	Edit Delete
2	Task 2 Title	This is the detail of task 2. lalalalala	asdf	important	Edit Delete

Task



Task

TaskTitle: TaskTitle

Detail: Detail

Level: serious

Principal: ---------

Task List

ID	Title	Detail	Responsible	Level	Operation
1	Task 1 Title	This is the detail of task 1	asdf	temporary	Edit Delete
2	Task 2 Title	This is the detail of task 2. lalalalala	asdf	important	Edit Delete
3	Task 3	This is task 3.	Jasmine	serious	Edit Delete

refresh

```
function bindBtnAddEvent() {
            $("#btnAdd").click(function () {
                $(".error-message").empty()

                $.ajax({
                    url: '/task/add/',
                    type: "post",
                    data: $("#formAdd").serialize(),
                    dataType: "JSON",
                    success: function (res) {
                        if (res.status) {
                            alert('Add Successful! ^-^');
                            // add reload
                            location.reload();
                        } else {
                            $.each(res.error, function (name, data) {
                                console.log(name, data);
                                $("#id_" + name).next().text(data[0])
                            })
                        }
                    }
                })
            })
        }
```

add page

```
def task_list(request):
    queryset = models.Task.objects.all().order_by('id')

    form = TaskModelForm()
    page_object = Pagination(request, queryset, page_size=2)

    context = {
        "form": form,
        "queryset": page_object.page_queryset,
        "page_string": page_object.html()
    }

    return render(request, 'task_list.html', context)
```

```
<nav aria-label="Page navigation">
            <ul class="pagination">
                {{ page_string }}
            </ul>
        </nav>
```