

---

# janta Documentation

*Release 2020*

**`__web_ghost__`**

**Nov 17, 2020**



**CONTENTS:**

<b>1</b>	<b>source</b>	<b>1</b>
1.1	ProjectJanta package . . . . .	1
1.2	janta package . . . . .	1
1.3	manage module . . . . .	16
<b>2</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>
	<b>Index</b>	<b>21</b>



## **1.1 ProjectJanta package**

### **1.1.1 Submodules**

### **1.1.2 ProjectJanta.asgi module**

### **1.1.3 ProjectJanta.settings module**

Django settings for ProjectJanta project.

Generated by ‘django-admin startproject’ using Django 3.1.3.

For more information on this file, see <https://docs.djangoproject.com/en/3.1/topics/settings/>

For the full list of settings and their values, see <https://docs.djangoproject.com/en/3.1/ref/settings/>

### **1.1.4 ProjectJanta.urls module**

### **1.1.5 ProjectJanta.wsgi module**

WSGI config for ProjectJanta project.

It exposes the WSGI callable as a module-level variable named `application`.

For more information on this file, see <https://docs.djangoproject.com/en/3.1/howto/deployment/wsgi/>

### **1.1.6 Module contents**

## **1.2 janta package**

### **1.2.1 Subpackages**

`janta.migrations` package

## Module contents

### 1.2.2 Submodules

#### 1.2.3 janta.admin module

#### 1.2.4 janta.apps module

```
class janta.apps.JantaConfig(app_name, app_module)
    Bases: django.apps.config.AppConfig

    name = 'janta'
```

#### 1.2.5 janta.forms module

```
class janta.forms.AddUserForm(data=None, files=None, auto_id='id_%s', prefix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=None, empty_permitted=False, field_order=None, use_required_attribute=None, renderer=None)
```

Bases: django.forms.forms.Form

Form for Add User

```
base_fields = {'regex': <django.forms.fields.CharField object>, 'user_handles': <dja
```

```
declared_fields = {'regex': <django.forms.fields.CharField object>, 'user_handles':
```

media

```
class janta.forms.ChangePasswordForm(data=None, files=None, auto_id='id_%s', prefix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=None, empty_permitted=False, field_order=None, use_required_attribute=None, renderer=None)
```

Bases: django.forms.forms.Form

Form for Change password

```
base_fields = {'conf_pass': <django.forms.fields.CharField object>, 'new_password':
```

```
declared_fields = {'conf_pass': <django.forms.fields.CharField object>, 'new_password'
```

media

```
class janta.forms.CreateTaskForm(data=None, files=None, auto_id='id_%s', prefix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=None, empty_permitted=False, field_order=None, use_required_attribute=None, renderer=None)
```

Bases: django.forms.forms.Form

Form for Create Task

```
base_fields = {'deadline': <django.forms.fields.DateField object>, 'description': <d
```

```
declared_fields = {'deadline': <django.forms.fields.DateField object>, 'description':
```

media

```
task_type_choices = [('Info', 'Info'), ('Text', 'Text input'), ('Upload', 'File upload
```

```
class janta.forms.DeleteTaskForm(data=None, files=None, auto_id='id_%s', pre-
                                fix=None, initial=None, error_class=<class
                                'django.forms.utils.ErrorList'>, label_suffix=None,
                                empty_permitted=False, field_order=None,
                                use_required_attribute=None, renderer=None)
```

Bases: django.forms.forms.Form

Form for Delete Task

```
base_fields = {'task_id': <django.forms.fields.CharField object>}
declared_fields = {'task_id': <django.forms.fields.CharField object>}
media
```

```
class janta.forms.EditProfileForm(data=None, files=None, auto_id='id_%s', pre-
                                fix=None, initial=None, error_class=<class
                                'django.forms.utils.ErrorList'>, label_suffix=None,
                                empty_permitted=False, field_order=None,
                                use_required_attribute=None, renderer=None)
```

Bases: django.forms.forms.Form

Form for Edit Profile

```
base_fields = {'about': <django.forms.fields.CharField object>, 'email': <django.form
declared_fields = {'about': <django.forms.fields.CharField object>, 'email': <django
gender_choices = [('Male', 'Male'), ('Female', 'Female')]
media
```

```
class janta.forms.LoginForm(data=None, files=None, auto_id='id_%s', prefix=None, ini-
                             tial=None, error_class=<class 'django.forms.utils.ErrorList'>,
                             label_suffix=None, empty_permitted=False, field_order=None,
                             use_required_attribute=None, renderer=None)
```

Bases: django.forms.forms.Form

Form For Login

```
base_fields = {'handle': <django.forms.fields.CharField object>, 'password': <django
declared_fields = {'handle': <django.forms.fields.CharField object>, 'password': <dj
media
```

```
class janta.forms.RegistrationForm(data=None, files=None, auto_id='id_%s', pre-
                                fix=None, initial=None, error_class=<class
                                'django.forms.utils.ErrorList'>, label_suffix=None,
                                empty_permitted=False, field_order=None,
                                use_required_attribute=None, renderer=None)
```

Bases: django.forms.forms.Form

Form for Registration

```
base_fields = {'conf_pass': <django.forms.fields.CharField object>, 'email': <django
declared_fields = {'conf_pass': <django.forms.fields.CharField object>, 'email': <dj
gender_choices = [('Male', 'Male'), ('Female', 'Female')]
media
user_type_choices = [('User', 'User'), ('Moderator', 'Moderator')]
```

```
class janta.forms.RemoveUserForm(data=None, files=None, auto_id='id_%s', pre-  
                                fix=None, initial=None, error_class=<class  
                                'django.forms.utils.ErrorList'>, label_suffix=None,  
                                empty_permitted=False, field_order=None,  
                                use_required_attribute=None, renderer=None)  
  
    Bases: django.forms.forms.Form  
    Form for Remove User  
  
    base_fields = {'user_handle': <django.forms.fields.CharField object>}  
    declared_fields = {'user_handle': <django.forms.fields.CharField object>}  
    media
```

```
class janta.forms.SendMessageForm(data=None, files=None, auto_id='id_%s', pre-  
                                fix=None, initial=None, error_class=<class  
                                'django.forms.utils.ErrorList'>, label_suffix=None,  
                                empty_permitted=False, field_order=None,  
                                use_required_attribute=None, renderer=None)  
  
    Bases: django.forms.forms.Form  
    Form for Send Message  
  
    base_fields = {'text': <django.forms.fields.CharField object>, 'user_handle': <django.forms.fields.CharField object>}  
    declared_fields = {'text': <django.forms.fields.CharField object>, 'user_handle': <django.forms.fields.CharField object>}  
    media
```

## 1.2.6 janta.models module

```
class janta.models.Messages  
    Bases: object  
    this class for Message Attributes:  
        id message id  
        sender message sender  
        receiver message receiver  
        text text of message  
        msg_time time of message  
        set_message (row)  
  
class janta.models.Notification  
    Bases: object  
    this class for get notifications  
        set_notification (row)  
  
class janta.models.Task  
    Bases: object  
    this class for task Attributes:  
        task_id task id  
        owner task owner
```



**task\_type** task type

**description** task description

**deadline** task deadline

**creation\_time** time of task creation

**set\_task** (*row*)

**class** janta.models.User

Bases: object

this class for User Attributes:

**user\_handle** user's handle

**user\_type** user's type

**user\_name** user's name

**user\_email** email address

**user\_gender** gender

**user\_about** user profile

**set\_user** (*row*)

**class** janta.models.UserModeratorMap

Bases: object

this class for User to Moderator Map Attributes: :user:user :moderator: moderator :map\_time: Mapping time

**set\_user\_moderator** (*row*)

**class** janta.models.UserTaskMap

Bases: object

this class for User to Task Map Attributes:

**task\_id** task id

**handle** handle which complete the task

**completion\_time** task completion time

:submission:task submitted or not :accepted: task accepted or not

**set\_user\_task\_map** (*row*)

janta.models.**accept\_submission** (*task\_id*, *handle*)

accept submission

janta.models.**add\_users** (*moderator*, *user\_handles*)

add user

janta.models.**admin\_del\_moderator** (*handle*)

delete Moderator

janta.models.**admin\_del\_task** (*task\_id*)

delete task for task\_id

janta.models.**admin\_del\_user** (*handle*)

delete user

```
janta.models.admin_get_task_list()  
    return Task List  
  
janta.models.admin_get_usr_modr_list()  
    return User Moderator list  
  
janta.models.change_password(handle, old_password, new_password)  
    change password  
  
janta.models.delete_message(msg_id, handle)  
    delete message  
  
janta.models.delete_task(task_id)  
    delete task  
  
janta.models.get_added_user(user, moderator)  
    get added user  
  
janta.models.get_messages(handle)  
    get message  
  
janta.models.get_my_tasks(owner, recent=False)  
    get my task  
  
janta.models.get_not_accepted_submissions(task_id)  
    get not accepted submissions  
  
janta.models.get_notifications(handle)  
    get unread notifications  
  
janta.models.get_ongoing_tasks(handle)  
    get ongoing tasks  
  
janta.models.get_overdue_tasks(handle)  
    get overdue tasks  
  
janta.models.get_recent_ongoing_tasks(handle)  
    get recent ongoing tasks  
  
janta.models.get_submission_status(task_id, moderator)  
    get submission status  
  
janta.models.get_submitted_tasks(handle)  
    get submitted tasks  
  
janta.models.get_task(task_id)  
    get task  
  
janta.models.get_user(handle)  
    get user  
  
janta.models.get_user_by_type(handle, user_type)  
    get user by type  
  
janta.models.get_user_task_map(task_id, handle)  
    get user task  
  
janta.models.insert_task(task_id, owner, description, task_type, deadline)  
    insert task  
  
janta.models.mark_as_read(noti_id, handle)  
    update the notifications mark as read
```

`janta.models.modify_task(task_id, owner, description, task_type, deadline)`  
modify task

`janta.models.notify_mapped_users(moderator, noti_type, text)`  
send notification to all mapped user

`janta.models.profile_edit(handle, name, emailid, gender, about, git_link, linkedin_link)`  
Edit Profile

`janta.models.reject_submission(task_id, handle)`  
reject submission

`janta.models.remove_user(user, moderator)`  
remove user

`janta.models.reset_db()`

`janta.models.send_message(sender, receiver, text)`  
send message

`janta.models.sent_notification(handle, noti_type, text)`  
send notification

`janta.models.submit_task(task_id, handle, submission)`  
submit task

`janta.models.user_register(handle, password, user_type, name, emailid, gender)`  
User Register

`janta.models.validate_credentials(handle, password)`  
validate Credentials

### 1.2.7 janta.mydb module

`janta.mydb.create_schema()`  
" create Schema

`janta.mydb.dictfetchall(cursor)`  
Return all rows from a cursor as a dict

`janta.mydb.execute_query(stmt, params)`  
" execute query and return this row

`janta.mydb.execute_update(stmt, params)`  
" update query and return this row

`janta.mydb.namedtuplefetchall(cursor)`  
Return all rows from a cursor as a namedtuple

### 1.2.8 janta.tests module

### 1.2.9 janta.viewhelper module

`janta.viewhelper.convert_to_json(unread_count, notifications)`  
This function convert data into json formate.

#### Parameters

- **unread\_count** – no of unread notifications

- **notifications** – unread notifications

**Returns**

**response\_json** notifications in json

`janta.viewhelper.is_admin(user_type)`

This function check that user type is Admin or not.

**Parameters** **user\_type** – user type

**Returns**

**True** if Admin

**False** if not Admin

`janta.viewhelper.is_description(description)`

This function check that description is valid or not.

**Parameters** **description** – description

**Returns**

**True** if valid

**False** if not valid

`janta.viewhelper.is_email(email)`

This function check that email is valid or not.

**Parameters** **email** – user email

**Returns**

**True** if valid

**False** if not valid

`janta.viewhelper.is_female(gender)`

This function check that user is Female or not.

**Parameters** **gender** – user gender

**Returns**

**True** if Female

**False** if not Female

`janta.viewhelper.is_file_name(name)`

This function check that file name is valid or not.

**Parameters** **name** – file name

**Returns**

**True** if valid

**False** if not valid

`janta.viewhelper.is_gender(gender)`

This function check that user have valid gender or not.

**Parameters** **gender** – user gender

**Returns**

**True** if valid gender

**False** if not valid

`janta.viewhelper.is_handle(handle)`

This function check that handle is valid or not.

**Parameters** `handle` – handle of user

**Returns**

**True** if valid

**False** if not valid

`janta.viewhelper.is_info(task_type)`

This function check that task type is Info or not.

**Parameters** `task_type` – task type

**Returns**

**True** if Info

**False** if not Info

`janta.viewhelper.is_male(gender)`

This function check that user is male or not.

**Parameters** `gender` – user gender

**Returns**

**True** if Male

**False** if not Male

`janta.viewhelper.is_moderator(user_type)`

This function check that user type is Moderator or not.

**Parameters** `user_type` – user type

**Returns**

**True** if Moderator

**False** if not Moderator

`janta.viewhelper.is_name(name)`

This function check that name is valid or not.

**Parameters** `name` – name of user

**Returns**

**True** if valid

**False** if not valid

`janta.viewhelper.is_password(password)`

This function check that password is valid or not.

**Parameters** `password` – password of handle

**Returns**

**True** if valid

**False** if not valid

`janta.viewhelper.is_task_id(task_id)`

This function check that task id is valid or not.

**Parameters** `task_id` – task id

**Returns**

**True** if valid

**False** if not valid

`janta.viewhelper.is_task_type(task_type)`

This function check that task type is valid or not.

**Parameters** `task_type` – task type

**Returns**

**True** if valid

**False** if not valid

`janta.viewhelper.is_text(task_type)`

This function check that task type is Text or not.

**Parameters** `task_type` – task type

**Returns**

**True** if Text

**False** if not Text

`janta.viewhelper.is_upload(task_type)`

This function check that task type is Upload or not.

**Parameters** `task_type` – task type

**Returns**

**True** if Upload

**False** if not Upload

`janta.viewhelper.is_user(user_type)`

This function check that user type is User or not.

**Parameters** `user_type` – user type

**Returns**

**True** if User

**False** if not User

`janta.viewhelper.is_user_type(user_type)`

This function check that user\_type is valid or not.

**Parameters** `user_type` – user type

**Returns**

**True** if valid

**False** if not valid

`janta.viewhelper.save_file(file, handle, task_id)`

This function save the file.

**Parameters**

- **file** – detail of the file
- **handle** – handle of user
- **task\_id** – task id

**Returns** return the filename

`janta.viewhelper.set_session(request, user)`

This function set the session for user.

**Parameters**

- **request** (*http request*) – the request sent for the session
- **user** – user details

**Returns** redirecting to the page according to http request

`janta.viewhelper.validate_add_user(form_data, moderator)`

This function validate the add\_user details.

**Parameters**

- **form\_data** – detail of the user's which are adding in this moderator
- **moderator** – moderator details

**Returns**

**status** status

**message** message

**handle\_list** list of handle which are adding

`janta.viewhelper.validate_change_password(form_data)`

This function validate the change\_password details.

**Parameters** **form\_data** – user details

**Returns**

**status** status

**message** message

`janta.viewhelper.validate_create_task(form_data)`

This function validate the create\_task details.

**Parameters** **form\_data** – task details

**Returns**

**status** status

**message** message

`janta.viewhelper.validate_delete_task(moderator, form_data)`

This function validate the delete\_task details.

**Parameters**

- **form\_data** – detail of task
- **moderator** – moderator details

**Returns**

**status** status

**message** message

`janta.viewhelper.validate_login(handle, password)`

This function validate the login Details.

**Parameters**

- **handle** – handle of user
- **password** – user password

**Returns**

**Status** status

**Message** message

`janta.viewhelper.validate_modify_task(form_data)`

This function validate the modify\_task details.

**Parameters** **form\_data** – task details

**Returns**

**Status** status

**Message** message

`janta.viewhelper.validate_profile_edit(form_data)`

This function validate the profile\_edit details.

**Parameters** **form\_data** – user details

**Returns**

**status** status

**message** message

`janta.viewhelper.validate_registration(form_data)`

This function validate the registration details.

**Parameters** **form\_data** – user details

**Returns**

**status** status

**message** message

`janta.viewhelper.validate_remove_user(form_data, moderator)`

This function validate the remove\_user details.

**Parameters**

- **form\_data** – detail of the user which are removeing in this moderator
- **moderator** – moderator details

**Returns**

**status** status

**message** message

**userhandle** handle of user



`janta.viewhelper.validate_send_message(form_data)`

This function validate the send\_message details.

**Parameters** `form_data` – message details

**Returns**

**status** status after send message

**message** message after send message

## 1.2.10 janta.views module

Create your views here.

This program deals with task function in our Janta application. Different functions related to task will be handled.

`janta.views.accept_submission(request)`

This function accepts the submitted task by request.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.add_users(request)`

This function add users by request.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.admin(request)`

This function maintains the admin page.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.admin_del_moderator(request)`

This function is used to delete the moderator by admin.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.admin_del_task(request)`

This function is used to delete the tasks by admin.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.admin_del_user(request)`

This function is used to delete the user by admin.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.change_password(request)`

This function is use to change password for the user.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.create_task(request)`

This function creates the task by request.

**type request** http request

**param request** the request sent for the task

**return** redirecting to the page according to http request

`janta.views.delete_messages(request)`

This function is use to delete the messages.

**Parameters request** (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.delete_task(request)`

This function deletes the task by request.

**Parameters request** (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.edit_profile(request)`

This function is use to edit the profile page of the user.

**Parameters request** (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.get_messages(request)`

This function is use to get the messages.

**type request** http request

**param request** the request sent for the task

**return** redirecting to the page according to http request

`janta.views.get_notifications(request)`

This function is use to get the notifications.

**Parameters request** (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.index(request)`

This function is the index page of user.

**Parameters request** (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.login(request)`

This function deals with the login of user by request.

**Parameters request** (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.logout(request)`

This function deals with the logout of the user by request.

**Parameters request** (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.mark_as_read(request)`

This function is used to read the messages and mark them as read.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.moderator_home(request)`

This function moderates the form request.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.modify_task(request)`

This function modifies the task by request.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.my_tasks(request)`

This function handles the functionality of the task by request.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.ongoing_tasks(request)`

This function maintains the ongoing task by request.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.overdue_tasks(request)`

This function maintains the overdue task by request.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.profile(request)`

This function is the profile page of the user.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.register(request)`

This function registers the user by request.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.reject_submission(request)`

This function rejects the submitted task by request.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.remove_user(request)`

This function removes the user by request.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.reset_database(request)`

This function is use to reset the database by request.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.send_message(request)`

This function is use to send the messages.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.submit_task(request)`

This function submits the task by request.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.submitted_tasks(request)`

This function keeps track of the submitted task by request.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.task_status(request)`

This function keeps track of the status of task by request.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

`janta.views.user_home(request)`

This function handles the notification by request.

**Parameters** `request` (*http request*) – the request sent for the task

**Returns** redirecting to the page according to http request

## 1.2.11 Module contents

## 1.3 manage module

Django's command-line utility for administrative tasks.

`manage.main()`

Run administrative tasks.

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### j

- janta, [16](#)
- janta.admin, [2](#)
- janta.apps, [2](#)
- janta.forms, [2](#)
- janta.migrations, [2](#)
- janta.models, [4](#)
- janta.mydb, [7](#)
- janta.tests, [7](#)
- janta.viewhelper, [7](#)
- janta.views, [13](#)

### m

- manage, [16](#)

### p

- ProjectJanta, [1](#)
- ProjectJanta.settings, [1](#)
- ProjectJanta.wsgi, [1](#)





## A

accept\_submission() (in module janta.models), 5  
 accept\_submission() (in module janta.views), 13  
 add\_users() (in module janta.models), 5  
 add\_users() (in module janta.views), 13  
 AddUserForm (class in janta.forms), 2  
 admin() (in module janta.views), 13  
 admin\_del\_moderator() (in module janta.models), 5  
 admin\_del\_moderator() (in module janta.views), 13  
 admin\_del\_task() (in module janta.models), 5  
 admin\_del\_task() (in module janta.views), 13  
 admin\_del\_user() (in module janta.models), 5  
 admin\_del\_user() (in module janta.views), 13  
 admin\_get\_task\_list() (in module janta.models), 5  
 admin\_get\_usr\_modr\_list() (in module janta.models), 6

## B

base\_fields (janta.forms.AddUserForm attribute), 2  
 base\_fields (janta.forms.ChangePasswordForm attribute), 2  
 base\_fields (janta.forms.CreateTaskForm attribute), 2  
 base\_fields (janta.forms.DeleteTaskForm attribute), 3  
 base\_fields (janta.forms.EditProfileForm attribute), 3  
 base\_fields (janta.forms.LoginForm attribute), 3  
 base\_fields (janta.forms.RegistrationForm attribute), 3  
 base\_fields (janta.forms.RemoveUserForm attribute), 4  
 base\_fields (janta.forms.SendMessageForm attribute), 4

## C

change\_password() (in module janta.models), 6  
 change\_password() (in module janta.views), 13  
 ChangePasswordForm (class in janta.forms), 2

convert\_to\_json() (in module janta.viewhelper), 7  
 create\_schema() (in module janta.mydb), 7  
 create\_task() (in module janta.views), 13  
 CreateTaskForm (class in janta.forms), 2

## D

declared\_fields (janta.forms.AddUserForm attribute), 2  
 declared\_fields (janta.forms.ChangePasswordForm attribute), 2  
 declared\_fields (janta.forms.CreateTaskForm attribute), 2  
 declared\_fields (janta.forms.DeleteTaskForm attribute), 3  
 declared\_fields (janta.forms.EditProfileForm attribute), 3  
 declared\_fields (janta.forms.LoginForm attribute), 3  
 declared\_fields (janta.forms.RegistrationForm attribute), 3  
 declared\_fields (janta.forms.RemoveUserForm attribute), 4  
 declared\_fields (janta.forms.SendMessageForm attribute), 4  
 delete\_message() (in module janta.models), 6  
 delete\_messages() (in module janta.views), 14  
 delete\_task() (in module janta.models), 6  
 delete\_task() (in module janta.views), 14  
 DeleteTaskForm (class in janta.forms), 2  
 dictfetchall() (in module janta.mydb), 7

## E

edit\_profile() (in module janta.views), 14  
 EditProfileForm (class in janta.forms), 3  
 execute\_query() (in module janta.mydb), 7  
 execute\_update() (in module janta.mydb), 7

## G

gender\_choices (janta.forms.EditProfileForm attribute), 3  
 gender\_choices (janta.forms.RegistrationForm attribute), 3

`get_added_user()` (in module `janta.models`), 6  
`get_messages()` (in module `janta.models`), 6  
`get_messages()` (in module `janta.views`), 14  
`get_my_tasks()` (in module `janta.models`), 6  
`get_not_accepted_submissions()` (in module `janta.models`), 6  
`get_notifications()` (in module `janta.models`), 6  
`get_notifications()` (in module `janta.views`), 14  
`get_ongoing_tasks()` (in module `janta.models`), 6  
`get_overdue_tasks()` (in module `janta.models`), 6  
`get_recent_ongoing_tasks()` (in module `janta.models`), 6  
`get_submission_status()` (in module `janta.models`), 6  
`get_submitted_tasks()` (in module `janta.models`), 6  
`get_task()` (in module `janta.models`), 6  
`get_user()` (in module `janta.models`), 6  
`get_user_by_type()` (in module `janta.models`), 6  
`get_user_task_map()` (in module `janta.models`), 6

## I

`index()` (in module `janta.views`), 14  
`insert_task()` (in module `janta.models`), 6  
`is_admin()` (in module `janta.viewhelper`), 8  
`is_description()` (in module `janta.viewhelper`), 8  
`is_email()` (in module `janta.viewhelper`), 8  
`is_female()` (in module `janta.viewhelper`), 8  
`is_file_name()` (in module `janta.viewhelper`), 8  
`is_gender()` (in module `janta.viewhelper`), 8  
`is_handle()` (in module `janta.viewhelper`), 9  
`is_info()` (in module `janta.viewhelper`), 9  
`is_male()` (in module `janta.viewhelper`), 9  
`is_moderator()` (in module `janta.viewhelper`), 9  
`is_name()` (in module `janta.viewhelper`), 9  
`is_password()` (in module `janta.viewhelper`), 9  
`is_task_id()` (in module `janta.viewhelper`), 9  
`is_task_type()` (in module `janta.viewhelper`), 10  
`is_text()` (in module `janta.viewhelper`), 10  
`is_upload()` (in module `janta.viewhelper`), 10  
`is_user()` (in module `janta.viewhelper`), 10  
`is_user_type()` (in module `janta.viewhelper`), 10

## J

`janta` (module), 16  
`janta.admin` (module), 2  
`janta.apps` (module), 2  
`janta.forms` (module), 2  
`janta.migrations` (module), 2  
`janta.models` (module), 4  
`janta.mydb` (module), 7  
`janta.tests` (module), 7  
`janta.viewhelper` (module), 7  
`janta.views` (module), 13

`JantaConfig` (class in `janta.apps`), 2

## L

`login()` (in module `janta.views`), 14  
`LoginForm` (class in `janta.forms`), 3  
`logout()` (in module `janta.views`), 14

## M

`main()` (in module `manage`), 16  
`manage` (module), 16  
`mark_as_read()` (in module `janta.models`), 6  
`mark_as_read()` (in module `janta.views`), 14  
`media` (`janta.forms.AddUserForm` attribute), 2  
`media` (`janta.forms.ChangePasswordForm` attribute), 2  
`media` (`janta.forms.CreateTaskForm` attribute), 2  
`media` (`janta.forms.DeleteTaskForm` attribute), 3  
`media` (`janta.forms.EditProfileForm` attribute), 3  
`media` (`janta.forms.LoginForm` attribute), 3  
`media` (`janta.forms.RegistrationForm` attribute), 3  
`media` (`janta.forms.RemoveUserForm` attribute), 4  
`media` (`janta.forms.SendMessageForm` attribute), 4  
`Messages` (class in `janta.models`), 4  
`moderator_home()` (in module `janta.views`), 15  
`modify_task()` (in module `janta.models`), 6  
`modify_task()` (in module `janta.views`), 15  
`my_tasks()` (in module `janta.views`), 15

## N

`name` (`janta.apps.JantaConfig` attribute), 2  
`namedtuplefetchall()` (in module `janta.mydb`), 7  
`Notification` (class in `janta.models`), 4  
`notify_mapped_users()` (in module `janta.models`), 7

## O

`ongoing_tasks()` (in module `janta.views`), 15  
`overdue_tasks()` (in module `janta.views`), 15

## P

`profile()` (in module `janta.views`), 15  
`profile_edit()` (in module `janta.models`), 7  
`ProjectJanta` (module), 1  
`ProjectJanta.settings` (module), 1  
`ProjectJanta.wsgi` (module), 1

## R

`register()` (in module `janta.views`), 15  
`RegistrationForm` (class in `janta.forms`), 3  
`reject_submission()` (in module `janta.models`), 7  
`reject_submission()` (in module `janta.views`), 15  
`remove_user()` (in module `janta.models`), 7  
`remove_user()` (in module `janta.views`), 15  
`RemoveUserForm` (class in `janta.forms`), 3

`reset_database()` (in module `janta.views`), 16  
`reset_db()` (in module `janta.models`), 7

## S

`save_file()` (in module `janta.viewhelper`), 10  
`send_message()` (in module `janta.models`), 7  
`send_message()` (in module `janta.views`), 16  
`SendMessageForm` (class in `janta.forms`), 4  
`sent_notification()` (in module `janta.models`), 7  
`set_message()` (`janta.models.Messages` method), 4  
`set_notification()` (`janta.models.Notification` method), 4  
`set_session()` (in module `janta.viewhelper`), 11  
`set_task()` (`janta.models.Task` method), 5  
`set_user()` (`janta.models.User` method), 5  
`set_user_moderator()` (`janta.models.UserModeratorMap` method), 5  
`set_user_task_map()` (`janta.models.UserTaskMap` method), 5  
`submit_task()` (in module `janta.models`), 7  
`submit_task()` (in module `janta.views`), 16  
`submitted_tasks()` (in module `janta.views`), 16

## T

`Task` (class in `janta.models`), 4  
`task_status()` (in module `janta.views`), 16  
`task_type_choices` (`janta.forms.CreateTaskForm` attribute), 2

## U

`User` (class in `janta.models`), 5  
`user_home()` (in module `janta.views`), 16  
`user_register()` (in module `janta.models`), 7  
`user_type_choices` (`janta.forms.RegistrationForm` attribute), 3  
`UserModeratorMap` (class in `janta.models`), 5  
`UserTaskMap` (class in `janta.models`), 5

## V

`validate_add_user()` (in module `janta.viewhelper`), 11  
`validate_change_password()` (in module `janta.viewhelper`), 11  
`validate_create_task()` (in module `janta.viewhelper`), 11  
`validate_credentials()` (in module `janta.models`), 7  
`validate_delete_task()` (in module `janta.viewhelper`), 11  
`validate_login()` (in module `janta.viewhelper`), 12  
`validate_modify_task()` (in module `janta.viewhelper`), 12  
`validate_profile_edit()` (in module `janta.viewhelper`), 12

`validate_registration()` (in module `janta.viewhelper`), 12  
`validate_remove_user()` (in module `janta.viewhelper`), 12  
`validate_send_message()` (in module `janta.viewhelper`), 12