

# Revised Alert Notice Application

Peter Higgins, PhD

April 18, 2022

## Introduction

This work is a revision of previous versions, and includes changes requested by AAVSO and other improvements thought reasonable. These are shown below:

1. (requested) The star table in the generated html alert notice has been reformatted to display RA and DEC values into two single strings without units. The units for them are shown in the table header. If a leading 0 is needed for a value such as 3 degrees it is added to show 03 degrees, and DEC values show '+/-'. Additionally, the precision for RA seconds is shown in hundreds of a second while the precision for DEC seconds is reduced to a tenth of a second. (see Figure 14)
2. (requested) Automate the upload of the notices' (optional) included table or graphic when the submission form is saved. No user action is required except to select a file. The input control has been changed to a file search control so that the user can select this file from the file manager. (see Figure 3)
3. (requested) Add AAVSO's forum page as a clickable link in the generated alert notice html document. (see Figure 14)

4. (requested) Verify save of feedback so it is shown in edit form.
5. Reformat appearance of the table lists of alert notices for both staff and non-staff views.
6. Restriction limiting generation of alert notices (HTML) to only those that have been 'approved' has been removed so staff can edit them as a draft document and save the draft separately.
7. Action buttons on Submit and Edit Submit forms have been moved to the top of these forms.
8. Non-staff submitters can edit only their last submittal and can not change previous submittals because these may already be processed by staff (see text for figure 20).
9. Non-staff submitters can not delete any submittal, even their own, now they must request deletes from staff.
10. Non-staff users can not generate an alert notice.

Both Principal Investigators, PI's and AAVSO staff can originate alert notices, but are allowed different privileges for editing, deleting and generating the alert notice document. These restrictions depend on the attributes of the user upon login.

### **Log in as staff member**

Logging in as a staff member directs the user to the page shown in Figure 1 which lists all alert notice records currently in the database regardless of submitter.



## Alert notices

Welcome peter

submit new alert

edit/view any alert

delete

alert html

logout

<input type="checkbox"/>	<a href="#">Notice Title</a>	<a href="#">Annum</a>	<a href="#">Submitted</a>	<a href="#">Lname</a>	<a href="#">Startdate</a>	<a href="#">Enddate</a>	<a href="#">Status</a>
<input type="checkbox"/>	test alert message	—	Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	accepted
<input type="checkbox"/>	test alert message	—	Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	accepted
<input type="checkbox"/>	test alert message two		Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert edited and accepted 1103		Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	accepted
<input type="checkbox"/>	test alert message pi		Dec. 1, 2021	Smith	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message pi edit		Dec. 1, 2021	Smith	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message		Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message		Dec. 1, 2021	smith	Dec. 10, 2021	Dec. 30, 2021	pending

Figure 1

The user actions associated with the top action buttons are shown in Figure 2:

```
<input class = "button" type="submit" action="" name="screate" value="submit new alert">
<input class = "button" type="submit" action="" name="sview" value="edit/view any alert">
<input class = "button" type="submit"; action="" name="sdelete"; value="delete">
<input class = "button" type="submit"; action="" name="pdf"; value="alert html">
<input class = "button" type="submit"; action="" name="lout"; value="logout">
```

Figure 2

Looking at each action:

### Submit new alert (staff)

Clicking on the button "submit new alert" directs to this code in views.py:

```
if "screate" in request.POST:
    form = SubmitForm() #used in production, produces blank form
    #form = testform() #used for testing form with values
    return render(request, "message2.html", {'form': form})
```

This code initiates a submit form then renders a template message2.html loaded with this form to accept user notice fields of data. The top section of the form template message2 is shown as Figure 3.

The screenshot shows the 'SUBMISSION' form. At the top, there is a logo for 'aavso' and two buttons: 'home' and 'save'. Below these is the word 'SUBMISSION' in red. The form is titled 'Submitter and dates' in blue. It contains several input fields: 'Submitted' (12 / 01 / 2021), 'Start date' (12 / 10 / 2021), 'End date' (12 / 30 / 2021), 'Urgency' (radio buttons for 'normal' and 'immediate'), 'Title' (test alert message), 'Last name' (Higgins), 'First name' (Peter), 'Affiliation' (none), and 'Contact' (higgins41@icloud.com). Below these fields is an optional section titled 'Optional: Please enter the file name of a graphic to include, limit 1 MByte'. It includes a 'Browse...' button and the text 'plot1.000001.png'.

Figure 3

Here it is seen that:

- Action buttons have been moved to top, save button in red
- Button to get included graphic has been changed to type FileField allowing file manager selection (note the browse button-when a file is selected and loaded into the form variable, the selection is shown beside it).

**save (a new notice from staff view)**

Saving the new notice form calls "new" in views.py, Figure 4

```

if "new" in request.POST:
    print("in new")
    sform = SubmitForm(request.POST, request.FILES)
    et=sform.errors
    print("et",et)
    rqp = request.POST
    title = rqp.get('title');
    if sform.is_valid():
        handle_uploaded_file(request.FILES['table'])
        sform.save()
        k = 0
        poststars(request, k)
        submitmsg = "Thank you for your submission of the Alert Notice \
                    entitled %s.<br> Your submission is now being reviewed. \
                    If you have any further questions,<br> \
                    please email eowaagon@aavso.org." %title
        messages.success(request, mark_safe(submitmsg))

```

Figure 4

This code gets the user's input from the Submit form and prints errors which would invalidate the form, then gets the title string to add to the success message. If the form is valid, an upload file handler is called to save the optional include file to the app's "static" directory, the form is saved, and a success message is displayed. The function to upload the include file is:

```

def handle_uploaded_file(f):
    with open('myalert/static/'+f.name, 'wb+') as destination:
        for chunk in f.chunks():
            destination.write(chunk)

```

When this form is complete, its lower half is given in Figure 5:

### Cadence

Hourly ☒

Nightly ☒

Weekly ☐

Monthly ☐

Time Series ☐

### Variability

Cataclysmic Variables ☒

Eclipsing Binary ☐

Exoplanet ☐

High Energy ☐

Long Period Variables ☐

Short Period Pulsators ☐

Young Stellar Objects ☐

### Spectroscopy

☐ low  $\leq 1000$

☒ medium  $\leq 5000$

☐ high  $> 5000$

wavelengths

### Modes of Observing

photometers VISUAL ☒ CCD-CMOS ☐ DSLR ☐ PEP ☐

filters U ☒ B ☐ V ☐ R ☐ I ☐ u ☐ g ☐ r ☐ l ☐ z ☐ CV ☐ CR ☐

**Add target VSX star names only, program will look them up in VSX, name\* not found**

S1	VY Scl	11.8 V	18.6 V	23h 29m 0.47s	-29d 46m 46.0s
S3	V0442 Oph	13.45 V	16.4 CV	17h 32m 15.14s	-16d 15m 22.1s
S5	V0794 Aql	13.4 V	18.6 V	20h 17m 33.92s	-03d 39m 51.0s
S7	V1024 Cep	14.7 V	20.7 CV	05h 06m 48.28s	+83d 19m 23.3s
S9	V0380 Oph	14.3 V	19.0 CV	17h 50m 13.62s	+006d 05m 29.3s
S2	WX Ari	14.5 V	18.6 V	02h 47m 36.22s	+10d 35m 37.7s
S4	GS Pav	14.8 CV	18.6 CV	20h 08m 7.62s	-69d 48m 58.8s
S6	V0704 And	12.6 CV	19.5 CV	23h 44m 57.51s	+43d 31m 22.3s
S8	LM UMa	8.21 V	(0.06) V	09h 46m 0.63s	+45d 52m 13.1s
S10	V0425 Cas	14.4 V	18.0 V	23h 03m 46.66s	+53d 17m 14.9s

Figure 5

The star table at the bottom of Figure 5 shows that leading zeros are added when needed in the star data string, and that "+-" is explicitly appended to Declination. When saved, the new alert notice is added to the list and a success message appears at the top of the staff list as shown in Figure 6

- 
- Thank you for your submission of the Alert Notice entitled test alert message. Your submission is now being reviewed. If you have any further questions, please email [eowaagon@aavso.org](mailto:eowaagon@aavso.org).
- 



## Alert notices

Welcome peter

[submit new alert](#)

[edit/view any alert](#)

[delete](#)

[alert html](#)

[logout](#)

<input type="checkbox"/>	<u>Notice Title</u>	<u>Annum</u>	<u>Submitted</u>	<u>Lname</u>	<u>Startdate</u>	<u>Enddate</u>	<u>Status</u>
<input type="checkbox"/>	test alert message	—	Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	accepted
<input type="checkbox"/>	test alert message	—	Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	accepted
<input type="checkbox"/>	test alert message two		Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert edited and accepted 1103		Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	accepted
<input type="checkbox"/>	test alert message pi		Dec. 1, 2021	Smith	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message pi edit		Dec. 1, 2021	Smith	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message		Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message		Dec. 1, 2021	smith	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message		Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	pending

Figure 6

Examination of the database confirms that the include file and the star data have been stored in the database. The model.py field entry

```
"table = models.CharField(max_length=60,null=True,blank=True )
```

assures the include file will be saved in the database, while the form field for it in forms.py facilitates the upload.

### Edit/View any Alert

Clicking on the button "edit/view any alert" goes to sview in views.py as shown in below:

```

if "sview" in request.POST:
    pks = request.POST.getlist("selection")
    sid = int(pks[0])
    dt = submittal.objects.get(id=sid)
    form = makeeditform(dt)
    return render(request, "smessage.html", {'form': form})

```

Now instead of loading an empty form for defining a new alert notice the code makes a dictionary of the entries in the selected record then loads an edit form from this dictionary and renders the template "smessage.html", Figure 7.

**Submission Edit**

**Submitter and dates**

Submitted 12 / 01 / 2021	Start date 12 / 10 / 2021	End date 12 / 30 / 2021	Feedback <div style="border: 1px solid #ccc; height: 100px;"></div>
Urgency <input checked="" type="radio"/> normal <input type="radio"/> immediate	Status <input type="radio"/> pending <input type="radio"/> accepted <input type="radio"/> conditional <input type="radio"/> rejected	Title test alert message	
Last name Higgins	First name Peter	Affiliation none	
Contact higgins41@icloud.com	Notice number 	Approved mm / dd / yyyy	
Optional: Please enter the file name of a graphic to include, limit 1 MByte Browse... No file selected.			

Figure 7

since the logged-in user was identified as a staff member, the populated edit form shown in figure 7 includes fields not seen by a non-staff submitter such as: feedback, status, Notice number and approval date. At present, staff determines the Notice number, but this can be created by the app. It is noteworthy that an edit must always identify the include graphic even if it doesn't change. As in the original input form, the action buttons are located on the top of the form and the save button is red. There is a new button on this form,"email feedback" discussed later.



## Save edit

When the staff member saves their edit, the code in Figure 8 is called:

```
if "edit" in request.POST:
    pid = request.POST.get('id')
    rqp = request.POST
    sid = rqp.get('id');
    title = rqp.get('title');
    auth = permit(request,sid)
    print("edit auth",auth)
    if auth == 1:
        sform = EditForm(request.POST,request.FILES)
        if sform.is_valid():
            print("sform is valid")
            upld = request.FILES['table']
            handle_uploaded_file(request.FILES['table'])
            nextid = submittal.objects.order_by('pk').last().id + 1
            sform.save()
            poststars(request, 0)
            pid = nextid - 1
            submittal.objects.filter(id=pid).delete()
            submitmsg = "Thank you for your edited submission of the Alert Notice \
                        entitled %s.<br> Your edited submission is now being reviewed. \
                        If you have any further questions,<br> \
                        please email eowaagon@aavso.org." % title
            messages.success(request, mark_safe(submitmsg))
```

Figure 8

The code saving the selected record by a privileged staff member shows that it captures the selected record and its id, verifies the user has save permission, retrieves the edit record (its fields and files), calls the function "handle\_upload\_file" to store the file in the static directory needed to retrieve the included file for placing on the notice html, and deletes the original record.

The edited notice is added to the list with its parent deleted, and a success message for the edit is shown, Figure 9.

- 
- Thank you for your edited submission of the Alert Notice entitled test alert message.  
Your edited submission is now being reviewed. If you have any further questions,  
please email [eowaagon@aavso.org](mailto:eowaagon@aavso.org).
- 



## Alert notices

Welcome peter

[submit new alert](#)
[edit/view any alert](#)
[delete](#)
[alert html](#)
[logout](#)

<input type="checkbox"/>	<a href="#">Notice Title</a>	<a href="#">Annum</a>	<a href="#">Submitted</a>	<a href="#">Lname</a>	<a href="#">Startdate</a>	<a href="#">Enddate</a>	<a href="#">Status</a>
<input type="checkbox"/>	test alert message	—	Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	accepted
<input type="checkbox"/>	test alert message	—	Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	accepted
<input type="checkbox"/>	test alert message two		Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert edited and accepted 1103		Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	accepted
<input type="checkbox"/>	test alert message pi		Dec. 1, 2021	Smith	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message pi edit		Dec. 1, 2021	Smith	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message		Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message		Dec. 1, 2021	smith	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message	—	Dec. 1, 2021	Higgins	Dec. 10, 2021	Dec. 30, 2021	accepted

Figure 9

### Email Feedback

Clicking this button sends the contents of feedback to the originating PI to instruct them on form details needing more, or better, information such as instructions or justification. The code for the email is given in Figure 10:

```
if "email" in request.POST:
    rqp = request.POST
    emsg = rqp.get('feedback');
    print("in feedback, emsg", emsg)
    fcontact = rqp.get('contact');
    print("in feedback, fcontact", fcontact)
    ftitle = rqp.get('title');
    subj = "Problems with " + ftitle
    print("in feedback, subject", subj)
    fstaff = "higgins41@icloud.com" # "phstpete@gmail.com" # "eowaagen.aavso.org"
    send_mail(subj, emsg, fstaff, [fcontact], fail_silently=False)
```

Figure 10

When deployed in production, the email will be the PI's contact address. An example email is seen in Figure 11.

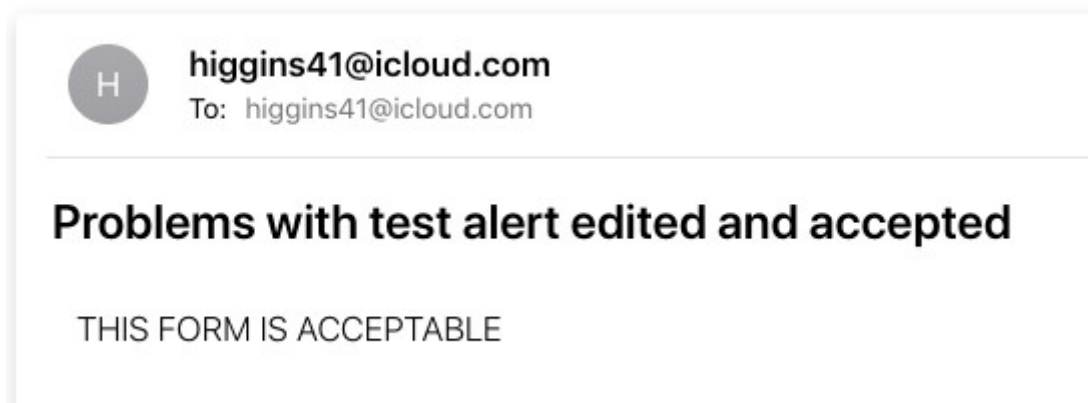


Figure 11

#### Alert notice document html

Clicking the alert html button shown in Figure 9 causes the app to run the code displayed in Figure 12:

```
if "pdf" in request.POST:
    print("in pdf")
    pks = request.POST.getlist("selection")
    print('pks', pks)
    sid = int(pks[0])
    print("sid", sid)
    dt = submittal.objects.get(id=sid)
    """
    if dt.status != 'accepted':
        submitmsg = "This message is not accepted"
        messages.success(request, submitmsg)
    else:
        name = release(request,sid)
        print("save alert html file",name)

        #return HttpResponseRedirect("Alert html created successfully")

        return FileResponse(open( name, 'rb'))
    """
    name = release(request,sid)
    print("save alert html file",name)
    return FileResponse(open( name, 'rb'))
```

Figure 12

The code between `"""` and `"""` has been commented out removing the restriction that an alert notice document can only be generated if the notice is approved. The code calls the function "release" that reads data from the a dictionary of the submission and writes the html document. A snippet from this writer code is shown below:

```
def release(request,rid):
    from datetime import date
    import PIL
    sb = submittal.objects.get(id=rid)

    today = date.today()
    fname = 'AN' + str(sb.annum) + '.html'
    f = open(fname, "w")
    f.write("<html>")
    f.write("<head>")
    f.write("<style>")
    f.write("table,th,td{border: none;}")
    f.write("</style>")
    f.write("</head>")
    f.write("<body>")

    full_name = "Attn AAVSO observers"
    f.write("<img src='static/aavso_logo.png' />")
```

Figure 13

A sample of the resulting html document is displayed as Figure 14.



**test alert message: number not assigned**

2022-04-17

Attn AAVSO observers

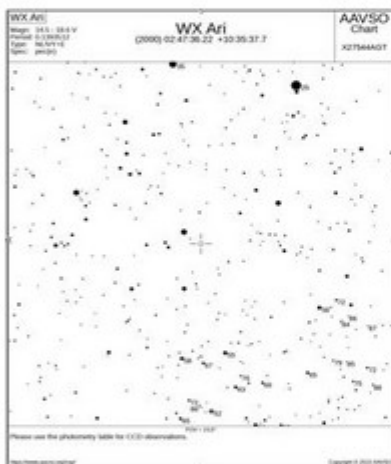
Colleagues:

An alert message entitled test alert message has been submitted by Peter Higgins with none on 2021-12-01 and was The justification given in this request is: testing.

The submitters instructions are: make observations

The submitters notes to observers follow: none

The submitters have included the following table:



Cadences selected are: hourly

Variability types indicated are:

Modes of Observing selected are:

Photometry filters that may be used include:

For further information, AAVSO forums: [Visit forum](#)

Target stars are (\* indicates not found in VSX):

Star	Con	MinV	MaxV	RA 2000 h:m:s	Dec 2000 d:m:s
VY	Scl	11.8	18.6	23 29 04.7	-29 46 46.0

Thank you very much for your support.

Sincerely,

AAVSO staff

Figure 14

Figure 14 shows the clickable link added to the forum page as well as the improved star table.

## **Log in as non-staff member such as University PI**

This app will appear to the user differently then described above when logged in as a non-staff user, such as a university PI, seeking ground observational support from AAVSO members by entering an alert notice. In this case the privileges extended to this user are limited to:

1. Viewing a list of all alert notices which they have submitted.
2. Authoring an original alert notice.
3. Editing, but not deleting, any alert notice having their email address.

How such a user is able to login in a production deployment of this app will have to be determined by the programmers who install this app in AAVSO's main website.

### **The login screen**

Upon login the main screen displayed is a list of all alert notices submitted by them as seen in Figure 15 below, which is restricted to only those notices with the loggedin user's (PI) email address.



## Alert notices

Welcome pi [Get account at AAVSO](#)

[submit new alert](#) [edit/view my alert](#) [logout](#)

<input type="checkbox"/>	<a href="#">Notice Title</a>	<a href="#">Annum</a>	<a href="#">Submitted</a>	<a href="#">Lname</a>	<a href="#">Startdate</a>	<a href="#">Enddate</a>	<a href="#">Status</a>
<input type="checkbox"/>	test alert message pi	Dec. 1, 2021	Smith	Dec. 10, 2021	Dec. 30, 2021	pending	
<input type="checkbox"/>	test alert message pi edit	Dec. 1, 2021	Smith	Dec. 10, 2021	Dec. 30, 2021	pending	
<input type="checkbox"/>	test alert message	Dec. 1, 2021	smith	Dec. 10, 2021	Dec. 30, 2021	pending	

Figure 15

Note that only two action buttons are present that allow creation and editing of new alerts while allowing viewing of any of their alert notices.

### Creation of a new alert notice

When a PI clicks the button for a new notice, the same form and template are used as was used by any staff member because the fields for status, feedback, and approval date are restricted to staff input in a staff edit. Figure 17 shows the code for new user input.

```

if "ucreate" in request.POST:
    print("usermenu/ucreate")
    try:
        print('in try')
        nextid = submittal.objects.order_by('pk').last().id + 1
        request.session['recnum'] = str(nextid)
    except:
        nextid = 1
        request.session['recnum'] = "1"
    usr = 0
    print("first init")
    #form = SubmitForm()
    form = testform()
    return render(request, "umessage.html", {'form': form})

```


Figure 17

umessage.html is actually the same as the template used for staff, but is another



template to allow for future programming of different input. As before, the code determines and stores the next record id. The rendered template is shown as Figure 18.

---

home save

  
**SUBMISSION**

**Submitter and dates**

Submitted <input type="text" value="12 / 01 / 2021"/>	Start date <input type="text" value="12 / 10 / 2021"/>	End date <input type="text" value="12 / 30 / 2021"/>	Urgency <input checked="" type="radio"/> normal <input type="radio"/> immediate	Title <input type="text" value="test alert message new"/>
Last name <input type="text" value="Smith"/>	First name <input type="text" value="Tom"/>	Affiliation <input type="text" value="none"/>	Contact <input type="text" value="phstpete@gmail.com"/>	

Optional: Please enter the file name of a graphic to include, limit 1 MByte

stars.jpg

Figure 18

Saving the new notice input by clicking the red "save" button follows the same save process as is described above for a staff member save. The new record is added to the list of previous alert notices for the same submitter, and a success message is displayed, Figure 19.

- Thank you for your submission of the Alert Notice entitled test alert message new. Your submission is now being reviewed. If you have any further questions, please email [eowaagon@aavso.org](mailto:eowaagon@aavso.org).



## Alert notices

Welcome pi [Get account at AAVSO](#)

<input type="checkbox"/>	<a href="#">Notice Title</a>	<a href="#">Annum</a>	<a href="#">Submitted</a>	<a href="#">Lname</a>	<a href="#">Startdate</a>	<a href="#">Enddate</a>	<a href="#">Status</a>
<input type="checkbox"/>	test alert message pi		Dec. 1, 2021	Smith	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message pi edit		Dec. 1, 2021	Smith	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message		Dec. 1, 2021	smith	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message new		Dec. 1, 2021	Smith	Dec. 10, 2021	Dec. 30, 2021	pending

Figure 19

### Clicking the edit/view button

Selecting a record from the list of notices, then clicking the edit/view button calls the function below, Figure 20.

```

if "uview" in request.POST:
    print("in view, user")
    pks = request.POST.getlist("selection")
    print('pks', pks)
    sid = int(pks[0])
    print("sid", sid)
    auth = permit(request, sid)
    skey=request.session['recnum']
    print('skey',skey)
    skey = int(request.session['recnum'])
    print("sid and session id", sid, skey)
    if auth == 1 and sid == skey:
        dt = submittal.objects.get(id=sid)
        print("dt.title", dt.title, dt.lname, dt.urgency, dt.table)
        form = makeeditform(dt)
        return render(request, "umessage.html", {'form': form})
    else:
        submitmsg = "you can only edit your last submittal"
        messages.success(request, submitmsg)

```

Figure 20

The requirement that equates sid to skey assures that the requested edit concerns the last submittal done by "ucreate" which stored the session variable 'recnum'. If the user requests edit of any other record this check rejects the edit request and tells the user with a message as seen in Figure 21.

- you can only edit your last submittal



## Alert notices

Welcome pi [Get account at AAVSO](#)

<input type="checkbox"/>	<u>Notice Title</u>	<u>Annum</u>	<u>Submitted</u>	<u>Lname</u>	<u>Startdate</u>	<u>Enddate</u>	<u>Status</u>
<input type="checkbox"/>	test alert message pi		Dec. 1, 2021	Smith	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message pi edit		Dec. 1, 2021	Smith	Dec. 10, 2021	Dec. 30, 2021	pending
<input type="checkbox"/>	test alert message		Dec. 1, 2021	smith	Dec. 10, 2021	Dec. 30, 2021	pending

Figure 21