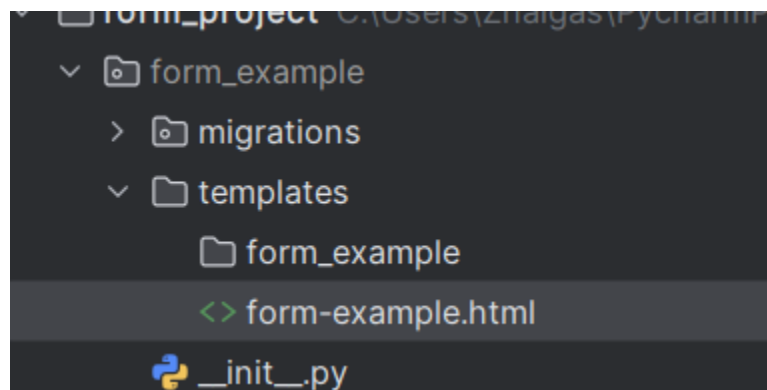


Exercise 6.01

```
C:\Users\Zhalgas\PycharmProjects>django-admin startproject form_project  
C:\Users\Zhalgas\PycharmProjects>cd form_project  
C:\Users\Zhalgas\PycharmProjects\form_project>python manage.py startapp form_example  
C:\Users\Zhalgas\PycharmProjects\form_project>
```



```
settings.py  form-example.html  views.py  urls.py  
2  <html lang="en">  
6  </head>  
7  <body>  
8      <form method="post">  
9          <p>  
10              <label for="id_text_input">Text Input</label><br>  
11              <input id="id_text_input" type="text"  
12                  name="text_input" value="" placeholder=  
13                  "Enter some text">  
14          </p>  
15          <p>  
16              <label for="id_password_input">Password Input  
17              </label><br>  
18              <input id="id_password_input" type="password"  
19                  name="password_input" value="" placeholder="Your  
20                  password">  
21          </p>  
22          <p>  
23              <input id="id_checkbox_input" type="checkbox"  
24                  name="checkbox_on" value="Checkbox Checked"  
25                  checked>  
26              <label for="id_checkbox_input">Checkbox</label>  
27          </p>  
28          <p>  
29              <input id="id_radio_one_input" type="radio"  
30                  name="radio_input" value="Value One">  
31              <label for="id_radio_one_input">Value One</label>  
32              <input id="id_radio_two_input" type="radio"  
33                  name="radio_input" value="Value Two" checked>  
34              <label for="id_radio_two_input">Value Two</label>  
35              <input id="id_radio_three_input" type="radio"  
36                  name="radio_input" value="Value Three">
```

```
def form_example(request):  
    return render(request, template_name: "form-example.html")
```

Text Input

Password Input

☒ Checkbox

☐ Value One ☒ Value Two ☐ Value Three

Favorite Book

Books You Own

Non-Fiction
Deep Learning with Keras
Web Development with Django
Fiction

Text Area

Number Input

Email Input

Date Input

Forbidden (403)

CSRF verification failed. Request aborted.

Help

Reason given for failure:

CSRF token missing.

In general, this can occur when there is a genuine Cross Site Request Forgery, or when [Django's CSRF mechanism](#) has not been used correctly.

- Your browser is accepting cookies.
- The view function passes a `request` to the template's [render](#) method.
- In the template, there is a `{% csrf_token %}` template tag inside each POST form that targets an internal URL.
- If you are not using `CsrfViewMiddleware`, then you must use `csrf_protect` on any views that use the `csrf_token` template tag, as well as the `csrf_token` view decorator.
- The form has a valid CSRF token. After logging in in another browser tab or hitting the back button after a login, you may need to refresh the page.

You're seeing the help section of this page because you have `DEBUG = True` in your Django settings file. Change that to `False`, and only the 403 error code will be shown.

You can customize this page using the `CSRF_FAILURE_VIEW` setting.

Exercise 6.02

Method: GET

Text Input

Password Input

☒ Checkbox

☐ Value One ☒ Value Two ☐ Value Three

Favorite Book

Books You Own

Non-Fiction

Deep Learning with Keras

Web Development with Django

Fiction

Text Area

Number Input

Email Input

Date Input



```

</head>
<body>
   <h4>Method: {{ method }}</h4>
  <form method="post">
    {% csrf_token %}
    <p>
      <label for="id_text_input">Text Input</

```

```

from django.shortcuts import render

def form_example(request):
    for name in request.POST:
        print("{}: {}".format(*args: name,
                                request.POST.getlist(name)))

    return render(request, template_name: "form-example.html", context: {"method": request.method})

```

Method: POST

```

[05/Mar/2025 18:31:06] "GET /form-example/ HTTP/1.1" 200 4278
csrfmiddlewaretoken: ['IV9azb7pCxe0WiY5z135WEFsuuAU6DayGe1lwmVT86xLKvLq8WU2tESL5i8bUKvn']
text_input: ['zhoma']
password_input: ['Zhoma1234!']
checkbox_on: ['Checkbox Checked']
radio_input: ['Value Two']
favorite_book: ['1']
books_you_own: ['1', '2']
text_area: ['']
number_input: ['']
email_input: ['']
date_input: ['2019-11-23']
submit_input: ['Submit Input']
hidden_input: ['Hidden Value']
[05/Mar/2025 18:32:14] "POST /form-example/ HTTP/1.1" 200 4279

```

settings.py form-example.html forms.py × views.py urls.py

```
1 from django import forms
2
3 RADIO_CHOICES = (
4     ("Value One", "Value One Display"),
5     ("Value Two", "Text For Value Two"),
6     ("Value Three", "Value Three's Display Text")
7 )
8
9 BOOK_CHOICES = (
10     (
11         "Non-Fiction", (
12             ("1", "Deep Learning with Keras"),
13             ("2", "Web Development with Django")
14         )
15     ),
16     (
17         "Fiction", (
18             ("3", "Brave New World"),
19             ("4", "The Great Gatsby")
20         )
21     )
22 )
23
```

```
class ExampleForm(forms.Form):
    text_input = forms.CharField()
    password_input = forms.CharField(widget=forms.PasswordInput)

    checkbox_on = forms.BooleanField()
    radio_input = forms.ChoiceField(choices=RADIO_CHOICES, widget=forms.RadioSelect)
    favorite_book = forms.ChoiceField(choices=BOOK_CHOICES)
    books_you_own = forms.MultipleChoiceField(choices=BOOK_CHOICES)
    text_area = forms.CharField(widget=forms.Textarea)
    integer_input = forms.IntegerField()
    float_input = forms.FloatField()
    decimal_input = forms.DecimalField()
    email_input = forms.EmailField()
    date_input = forms.DateField(widget=forms.DateInput(attrs = {"type": "date"}))
    hidden_input = forms.CharField(widget=forms.HiddenInput, initial = "Hidden Value")
```

```
settings.py  <> form-example.html  forms.py  views.py  urls.py

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6  </head>
7  <body>
8      <h4>Method: {{ method }}</h4>
9      <form method="post">
10         {% csrf_token %}
11         {{ form.as_p }}
12     <p>
13         <input type="submit" name="submit_input"
14         value="Submit Input">
15     </p>
16     <p>
17         <button type="submit" name="button_element"
18         value="Button Element">
19             Button With <strong>Styled</strong> Text
20         </button>
21     </p>
22 </form>
23 </body>
24 </html>
```

```
settings.py  <> form-example.html  forms.py  views.py  urls.py

1  from django.shortcuts import render
2  from forms import ExampleForm
3
4  def form_example(request):
5      form = ExampleForm()
6      for name in request.POST:
7          print("{}: {}".format(*args: name,
8                                request.POST.getlist(name)))
9
10 <> return render(request, template_name: "form-example.html", context: {"method": request.method, "form": form})
11
```

```
[05/Mar/2025 21:04:32] "GET / HTTP/1.1" 404 2319
[05/Mar/2025 21:04:39] "GET /form-example HTTP/1.1" 301 0
[05/Mar/2025 21:04:39] "GET /form-example/ HTTP/1.1" 200 3637
csrfmiddlewaretoken: ['PUoaiitf4iJJJC8vYmFdzk3Vl87aTXfFNdglfthJAR2GxPVQxhwa6kgeWWFrH4Au']
text_input: ['zhoma']
password_input: ['Zhoma1234!']
checkbox_on: ['on']
radio_input: ['Value Three']
favorite_book: ['1']
books_you_own: ['1', '2', '3', '4']
text_area: ['fsdfsdfsdfsdf']
integer_input: ['322']
float_input: ['432.4323']
decimal_input: ['432324.4234324234']
email_input: ['arystambekdimash005@gmail.com']
date_input: ['2025-03-20']
hidden_input: ['Hidden Value']
button_element: ['Button Element']
[05/Mar/2025 21:05:36] "POST /form-example/ HTTP/1.1" 200 3638
```


Method: POST

Text input:

Password input:

Checkbox on: ☐

Radio input:

- ☐ Value One Display
- ☐ Text For Value Two
- ☐ Value Three's Display Text

Favorite book:

Books you own:

Non-Fiction
Deep Learning with Keras
Web Development with Django

Fiction

Text area:

Integer input:

Float input:

Decimal input:

Email input:

Exercise 6.04

```
def form_example(request):
    if request.method == "POST":
        form = ExampleForm(request.POST)
        if form.is_valid():
            for name, value in form.cleaned_data.items():
                print("{}: ({}). {}".format(*args: name,
                                                type(value), value))
        else:
            form = ExampleForm()

    return render(request, template_name: "form-example.html", context: {"method": request.method, "form": form})
```

- Ensure that there are no more than 3 digits in total.

Decimal input:

```
[03/nar/2025-11-11:04] POST /form-example/ HTTP/1.1 200 3916
text_input: (<class 'str'>) zhoma
password_input: (<class 'str'>) Zhoma1234!
checkbox_on: (<class 'bool'>) True
radio_input: (<class 'str'>) Value Two
favorite_book: (<class 'str'>) 2
books_you_own: (<class 'list'>) ['1', '2']
text_area: (<class 'str'>) fdsfsf
integer_input: (<class 'int'>) 213213
float_input: (<class 'float'>) 31213.0
decimal_input: (<class 'decimal.Decimal'>) 321
email_input: (<class 'str'>) zhalgas.seidazym2005@gmail.com
date_input: (<class 'datetime.date'>) 2025-03-28
hidden_input: (<class 'str'>) Hidden Value
```

Exercise 6.05

```
class ExampleForm(forms.Form):
    text_input = forms.CharField(max_length=3)
    password_input = forms.CharField(min_length=8, widget=forms.PasswordInput)

    checkbox_on = forms.BooleanField()
    radio_input = forms.ChoiceField(choices=RADIO_CHOICES, widget=forms.RadioSelect)
    favorite_book = forms.ChoiceField(choices=BOOK_CHOICES)
    books_you_own = forms.MultipleChoiceField(required=False, choices=BOOK_CHOICES)
    text_area = forms.CharField(widget=forms.Textarea)
    integer_input = forms.IntegerField(min_value=1, max_value=10)
    float_input = forms.FloatField()
    decimal_input = forms.DecimalField(max_digits=5, decimal_places=3)
    email_input = forms.EmailField()
    date_input = forms.DateField(widget=forms.DateInput(attrs = {"type": "date"}))
    hidden_input = forms.CharField(widget=forms.HiddenInput, initial = "Hidden Value")
```

Integer input:

! Значение должно быть меньше или равно 10.

Decimal input:

Decimal input:

! Введите допустимое значение. Ближайшие: 431413,431 и 431413,432.

Date input:

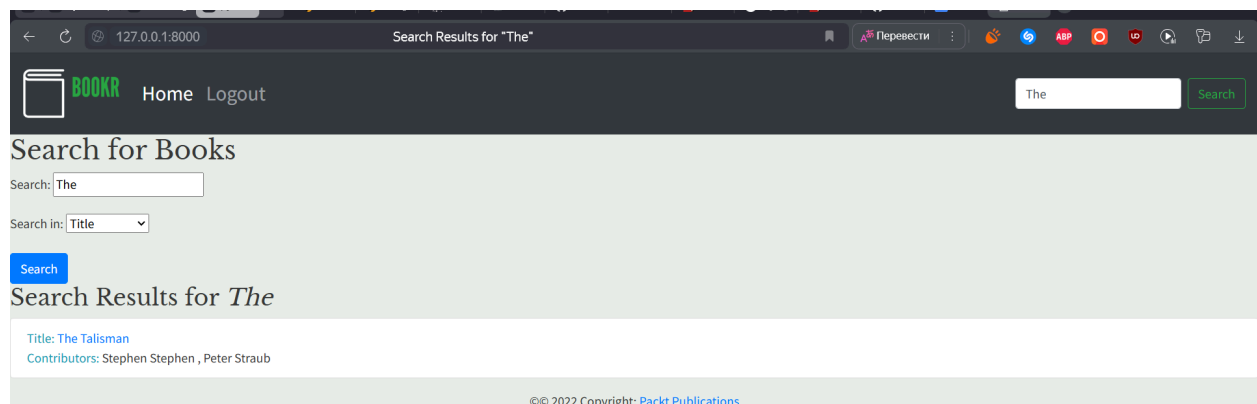
Password input:

! Минимальное количество символов: 8. Длина текста сейчас: 7.

Radio input:

```
checkbox_on: (<class 'bool'>) True
radio_input: (<class 'str'>) Value Three
favorite_book: (<class 'str'>) 1
books_you_own: (<class 'list'>) ['1']
text_area: (<class 'str'>) fsdfsd
integer_input: (<class 'int'>) 10
float_input: (<class 'float'>) 213.0
decimal_input: (<class 'decimal.Decimal'>) 32.12
email_input: (<class 'str'>) zhalgas.seidazym2005@gmail.com
date_input: (<class 'datetime.date'>) 2025-03-28
hidden_input: (<class 'str'>) Hidden Value
```

Activity 6.01



```

def book_search(request):
    search_text = request.GET.get("search", "")
    form = SearchForm(request.GET)
    books = set()

    if form.is_valid() and form.cleaned_data['search']:
        search = form.cleaned_data['search']
        search_in = form.cleaned_data['search_in'] or 'title'
        if search_in == 'title':
            books = Book.objects.filter(title__icontains=search)
        else:
            f_name = Contributor.objects.filter(first_names__icontains=search)

            for contributor in f_name:
                for book in contributor.book_set.all():
                    books.add(book)

            l_name = Contributor.objects.filter(last_names__icontains=search)

            for contributor in l_name:
                for book in contributor.book_set.all():
                    books.add(book)

    return render(request, template_name="reviews/search.html", context={"search_text": search_text, 'form': fo

```

```

<title>{% block title %}Bookr {% endblock title %}</title>
</head>

```

```

<form action="{% url 'book_search' %}" class="form-inline my-2 my-lg-0">
    <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search"
    name="search" value="{% search_text %}" minlength="3">
    <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
</form>

```

```

{% block content %}
    <h2>Search for Books</h2>

    <form>
        {{ form.as_p }}

        <button type="submit" class="btn btn-primary">Search</button>
    </form>
    {% if form.is_valid and search_text %}
        <h3>Search Results for <em>{{ search_text }}</em></h3>
        <ul class="list-group">
            {% for book in books %}
                <li class="list-group-item">
                    <span class="text-info">Title: </span>
                    <a href="{% url 'book_detail' book.pk %}">{{ book }}</a>
                    <br>
                    <span class="text-info">Contributors: </span>
                    {% for contributor in book.contributors.all %}
                        {{ contributor.first_names }} {{ contributor.last_names }}
                        {% if not forloop.last %}, {% endif %}
                    {% endfor %}
                </li>
                {% empty %}
                    <li class="list-group-item">No results found.</li>
            {% endfor %}
        </ul>
    {% endif %}

```

```

{% extends 'base.html' %}

{% block title %}
    {% if form.is_valid and search_text %}
        Search Results for "{{ search_text }}"
    {% else %}
        Book Search
    {% endif %}
{% endblock %}

{% block content %}

```

Exercise 7.01

```

def validate_email_domain(value):
    if value.split("@")[-1].lower() != "example.com":
        raise ValidationError("The email address must be on the domain example.com.")

class OrderForm(forms.Form):
    magazine_count = forms.IntegerField(min_value=0, max_value=80)
    book_count = forms.IntegerField(min_value=0, max_value=50)
    send_confirmation = forms.BooleanField(required=False)
    email = forms.EmailField(required=False, validators=[validate_email_domain])

    def clean_email(self):
        return self.cleaned_data['email'].lower()

    def clean(self):
        cleaned_data = super().clean()
        if cleaned_data["send_confirmation"] and not cleaned_data.get("email"):
            self.add_error(field="email", error="Please enter an email address to receive the confirmation message.")
        elif cleaned_data.get("email") and not cleaned_data["send_confirmation"]:
            self.add_error(field="send_confirmation",
                           error="Please check this if you want to receive a confirmation email.")
        item_total = cleaned_data.get("magazine_count", 0) + cleaned_data.get("book_count", 0)
        if item_total > 100:
            self.add_error(field=None, error="The total number of items must be 100 or less.")

```

```

from .forms import ExampleForm, OrderForm

def form_example(request):
    if request.method == "POST":
        form = OrderForm(request.POST)
        if form.is_valid():
            for name, value in form.cleaned_data.items():
                print("{}: ({} {})".format(*args: name,
                                           type(value), value))
        else:
            form = OrderForm()

    return render(request, template_name="form-example.html", context={"method": request.method, "form": form})

```

Method: POST

- The total number of items must be 100 or less.

Magazine count:

Book count:

- Please check this if you want to receive a confirmation email.

Send confirmation: ☐

Email:

Send confirmation: ☒

- The email address must be on the domain example.com.
- Please enter an email address to receive the confirmation message.

Email:

Exercise 7.02

Method: GET

Magazine count:

Book count:

Send confirmation: ☐

Email:

Exercise 7.03



Name: The name of the Publisher.

Website: The Publisher's website.

Email: The Publisher's email address.



Name: The name of the Publisher.

Website: The Publisher's website.

Email: The Publisher's email address.

```
settings.py  urls.py  models.py  forms.py  <> form-example.html x  view
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6  </head>
7  <body>
8      {% for message in messages %}
9      <p><em>{{ message.level_tag|title }}:</em> {{ message }}</p>
10     {% endfor %}
11     <form method="post">
12         {% csrf_token %}
13         {{ form.as_p }}
14         <p>
15             <input type="submit" value="Submit">
16         </p>
17     </form>
18 </body>
19 </html>
```

```
class PublisherForm(forms.ModelForm):
    class Meta:
        model = Publisher
        fields = "__all__"
```

```
def publisher_edit(request, pk=None):
    if pk is not None:
        publisher = get_object_or_404(Publisher, pk=pk)
    else:
        publisher = None

    if request.method == "POST":
        form = PublisherForm(request.POST, instance=publisher)
        if form.is_valid():
            updated_publisher = form.save()
            if publisher is None:
                messages.success(
                    request, 'Publisher "{}" was created.'.format(updated_publisher)
                )
            else:
                messages.success(
                    request, 'Publisher "{}" was updated.'.format(updated_publisher)
                )

            return redirect(to="publisher_edit", *args: updated_publisher.pk)
        else:
            form = PublisherForm(instance=publisher)

    return render(
        request, template_name="form-example.html", context={"method": request.method, "form": form}
    )
```

Activity 7.01

```

</div>
<div class="container-fluid">
  {% for message in messages %}
    <div class="alert alert-{% if message.level_tag == 'error' %}danger{% else %}{% message.level_tag %}"
      role="alert">
      {{ message }}
    </div>
  {% endfor %}

  {% block content %}
    <h1>Welcome to Bookr!</h1>
  {% endblock %}
</div>


```

```

return render(
    request,
    template_name="reviews/instance-form.html",
    context={
        "method": request.method,
        "form": form,
        "model_type": "Publisher",
        "instance": publisher,
    },
)

```

```
1 {% extends 'reviews/base.html' %}
2
3 {% block title %}
4     {% if instance %}
5         Editing {{ model_type }} {{ instance }}
6     {% else %}
7         New {{ model_type }}
8     {% endif %}
9 {% endblock title %}
10
11 {% block content %}
12     <h2>
13         {% if instance %}
14             Editing {{ model_type }} <em>{{ instance }}</em>
15         {% else %}
16             New {{ model_type }}
17         {% endif %}
18     </h2>
19
20     <form method="post">
21         {% csrf_token %}
22         {{ form.as_p }}
23         <button type="submit" class="btn btn-primary">
24             {% if instance %}Save{% else %}Create{% endif %}
25         </button>
26     </form>
27 {% endblock %}
```

 [Home](#) [Logout](#)

Editing Publisher *New Town Publisher*


Name: The name of the Publisher.

Website: The Publisher's website.

Email: The Publisher's email address.

©© 2022 Copyright: [Packt Publ](#)
Website bv: Packt Publicati

← ↻ 127.0.0.1:8000 New Publisher

 [Home](#) [Logout](#)

New Publisher

Name: The name of the Publisher.

Website: The Publisher's website.

Email: The Publisher's email address.

[Create](#)

Activity 7.02

```
class ReviewForm(forms.ModelForm):
    class Meta:
        model = Review
        exclude = ["date_edited", "book"]

    rating = forms.IntegerField(min_value=0, max_value=5)
```

```

def review_edit(request, book_pk, review_pk=None):
    book = get_object_or_404(Book, pk=book_pk)

    if review_pk is not None:
        review = get_object_or_404(Review, book_id=book_pk, pk=review_pk)
    else:
        review = None

    if request.method == "POST":
        form = ReviewForm(request.POST, instance=review)
        if form.is_valid():
            updated_review = form.save(False)
            updated_review.book = book

            if review is None:
                messages.success(request, 'Review for "{}" created.'.format(book))
            else:
                updated_review.date_edited = timezone.now()
                messages.success(request, 'Review for "{}" updated.'.format(book))

            updated_review.save()
            return redirect(to="book_detail", *args: book.pk)
        else:
            form = ReviewForm(instance=review)

    return render(
        request,
        template_name="reviews/instance-form.html",
        context={
            "form": form,
            "instance": review

```

```

path(route: "publishers/new/", views.publisher_edit, name="publisher_create"),
path(route: "books/<int:book_pk>/reviews/new/", views.review_edit, name="review_create"),
path(route: "books/<int:book_pk>/reviews/<int:review_pk>/", views.review_edit, name="review_edit",
),

```



```

12     <br>
13     {% if not reviews %}
14         <hr>
15         <h3>Be the first one to write a review.</h3>
16     {% else %}
17         <span class="text-info">Overall Rating: </span><span class="badge badge-primary badge-pill"
18         <br>
19         <hr>
20         <h3>Review Comments</h3>
21         <ul>
22             {% for review in reviews %}
23                 <li class="list-group-item">
24                     <span class="text-info">Review comment: </span> <span>{{ review.content }}</spa
25                     <br>
26                     <span class="text-info">Created on: </span><span>{{ review.date_created }}</spa
27                     <br>
28                     <span class="text-info">Modified on: </span><span>{{ review.date_edited }}</spa
29                     <br>
30                     <span class="text-info">Rating: </span><span class="badge badge-secondary badge
31                     <br>
32                     <span class="text-info">Creator: </span><span>{{ review.creator.username }}</sp
33                     <br>
34                 </li>
35                 <a href="{% url 'review_edit' book.pk review.pk %}">Edit Review</a>
36             {% endfor %}
37         </ul>
38     {% endif %}
39     <a class="btn btn-primary" href="{% url 'review_create' book.pk %}">Add Review</a>
40 {% endblock %}

```



Reviews

Review for "Advanced Deep Learning with Keras" created.

Book Details

Title: Advanced Deep Learning with Keras

Publisher: Packt Publishing

Publication Date: Oct. 31, 2018

Overall Rating: 4

Review Comments

Review comment: sdfsd

Created on: March 5, 2025, 7:25 p.m.

Modified on: None

Rating: 4

Creator: carol

[Edit Review](#)

[Add Review](#)

Reviews

Home

Logout

Editing Review *Review object (2)*

For Book *Advanced Deep Learning with Keras*

Content:

sdfsdf

The Review text.

Rating:

4

Creator:

carol

▼

Save

©© 20...

Project link
https://github.com/zhalgas-seidazym/event_manager



Review for "Advanced Deep Learning with Keras" updated.

Book Details

Title: Advanced Deep Learning with Keras

Publisher: Packt Publishing

Publication Date: Oct. 31, 2018

Overall Rating: 4

Review Comments

Review comment: sdfsd fsf

Created on: March 5, 2025, 7:25 p.m.

Modified on: March 5, 2025, 7:26 p.m.

Rating: 4

Creator: carol

[Edit Review](#)