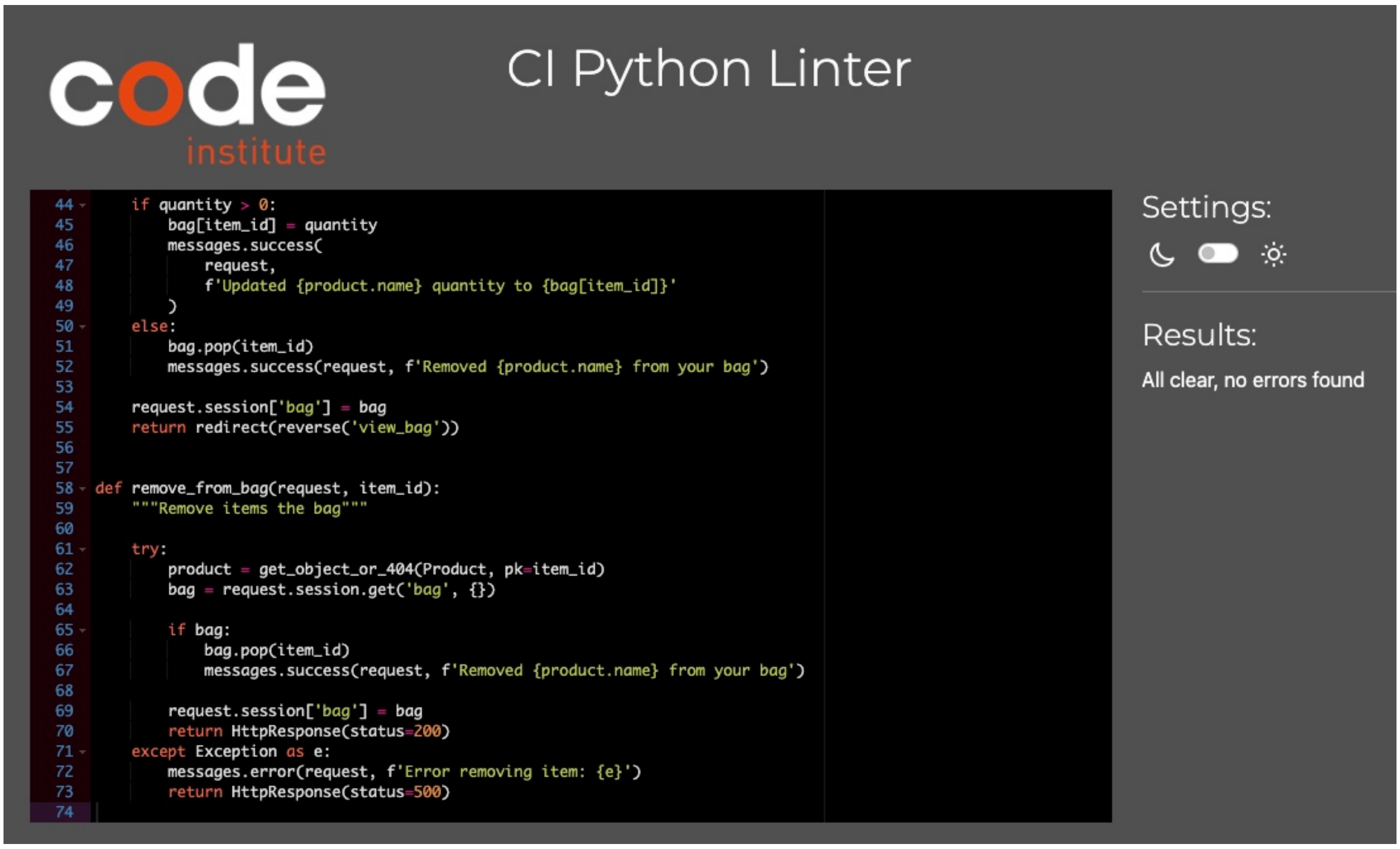
Bag app

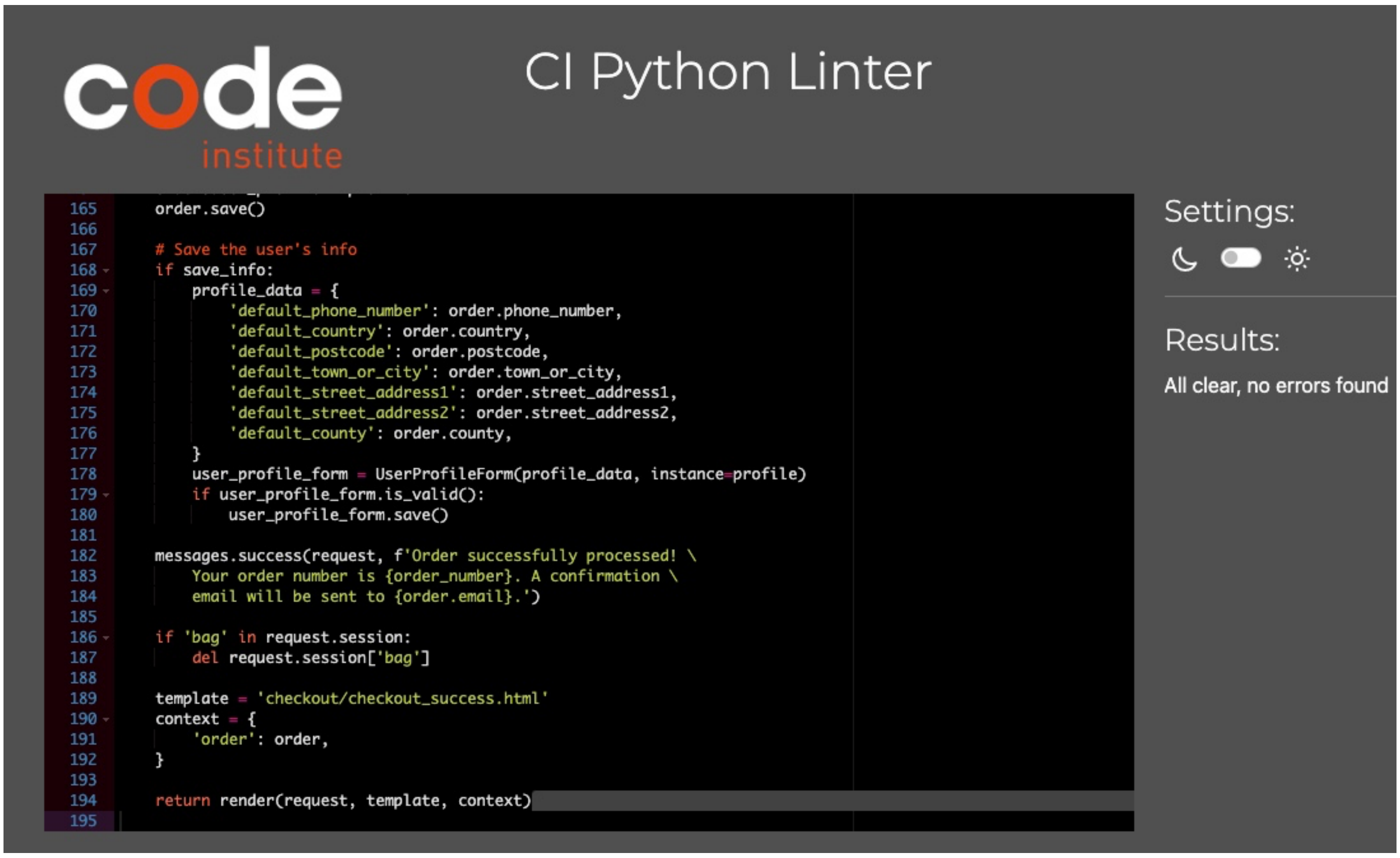


```
    if quantity > 0:
        bag[item_id] = quantity
        messages.success(
            request,
            f'Updated {product.name} quantity to {bag[item_id]}'
        )
    else:
        bag.pop(item_id)
        messages.success(request, f'Removed {product.name} from your bag')

    request.session['bag'] = bag
    return redirect(reverse('view_bag'))


def remove_from_bag(request, item_id):
    """Remove items the bag"""

    try:
        product = get_object_or_404(Product, pk=item_id)
        bag = request.session.get('bag', {})

        if bag:
            bag.pop(item_id)
            messages.success(request, f'Removed {product.name} from your bag')

        request.session['bag'] = bag
        return HttpResponse(status=200)
    except Exception as e:
        messages.error(request, f'Error removing item: {e}')
        return HttpResponse(status=500)
```

Results:

All clear, no errors found

Checkout app



CI Python Linter

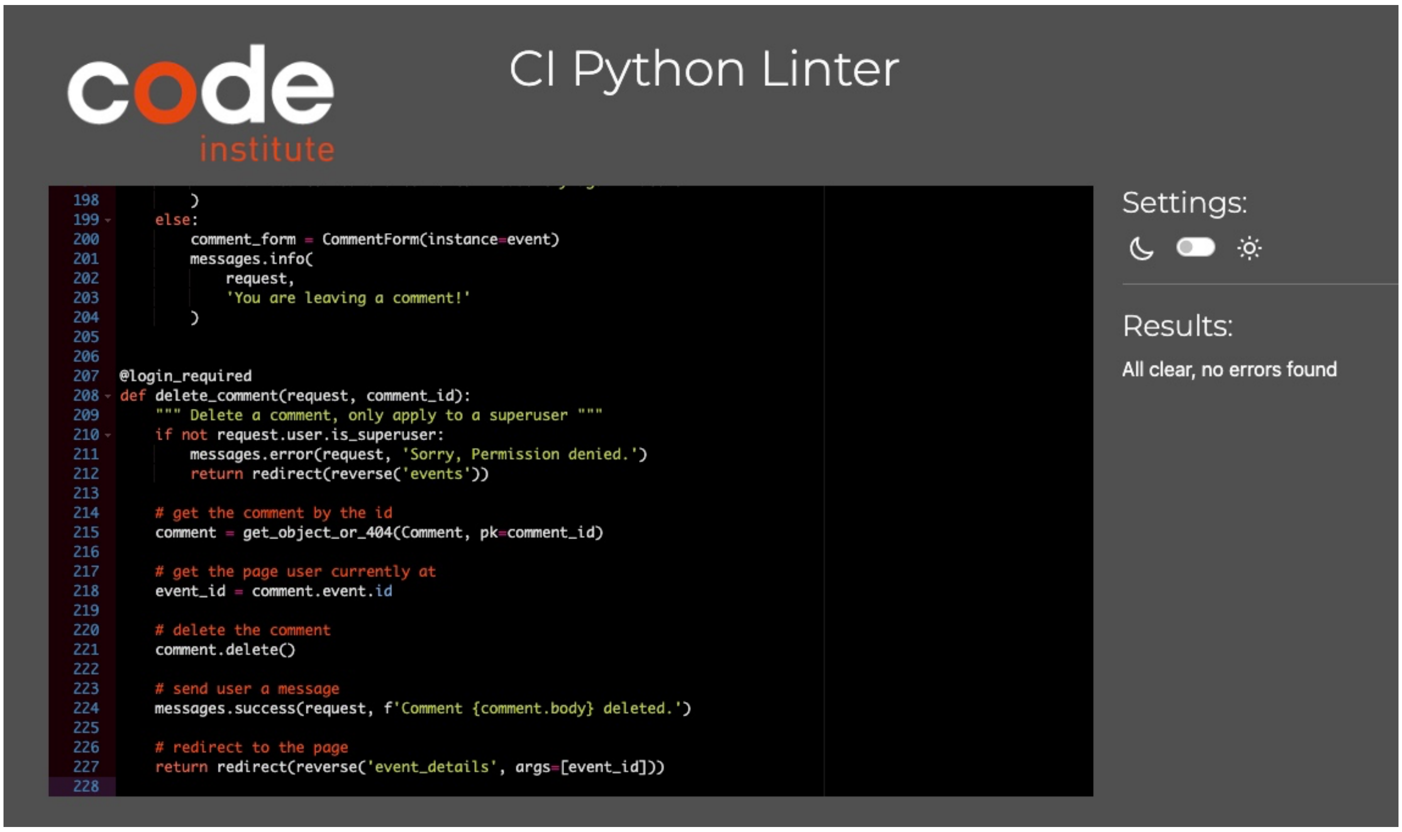
```
    order.save()

    # Save the user's info
    if save_info:
        profile_data = {
            'default_phone_number': order.phone_number,
            'default_country': order.country,
            'default_postcode': order.postcode,
            'default_town_or_city': order.town_or_city,
            'default_street_address1': order.street_address1,
            'default_street_address2': order.street_address2,
            'default_county': order.county,
        }
        user_profile_form = UserProfileForm(profile_data, instance=profile)
        if user_profile_form.is_valid():
            user_profile_form.save()

    messages.success(request, f'Order successfully processed! \
        Your order number is {order_number}. A confirmation \
        email will be sent to {order.email}.')

    if 'bag' in request.session:
        del request.session['bag']

    template = 'checkout/checkout_success.html'
    context = {
        'order': order,
    }

    return render(request, template, context)
```

Settings:

Results:

All clear, no errors found

Comment app



CI Python Linter

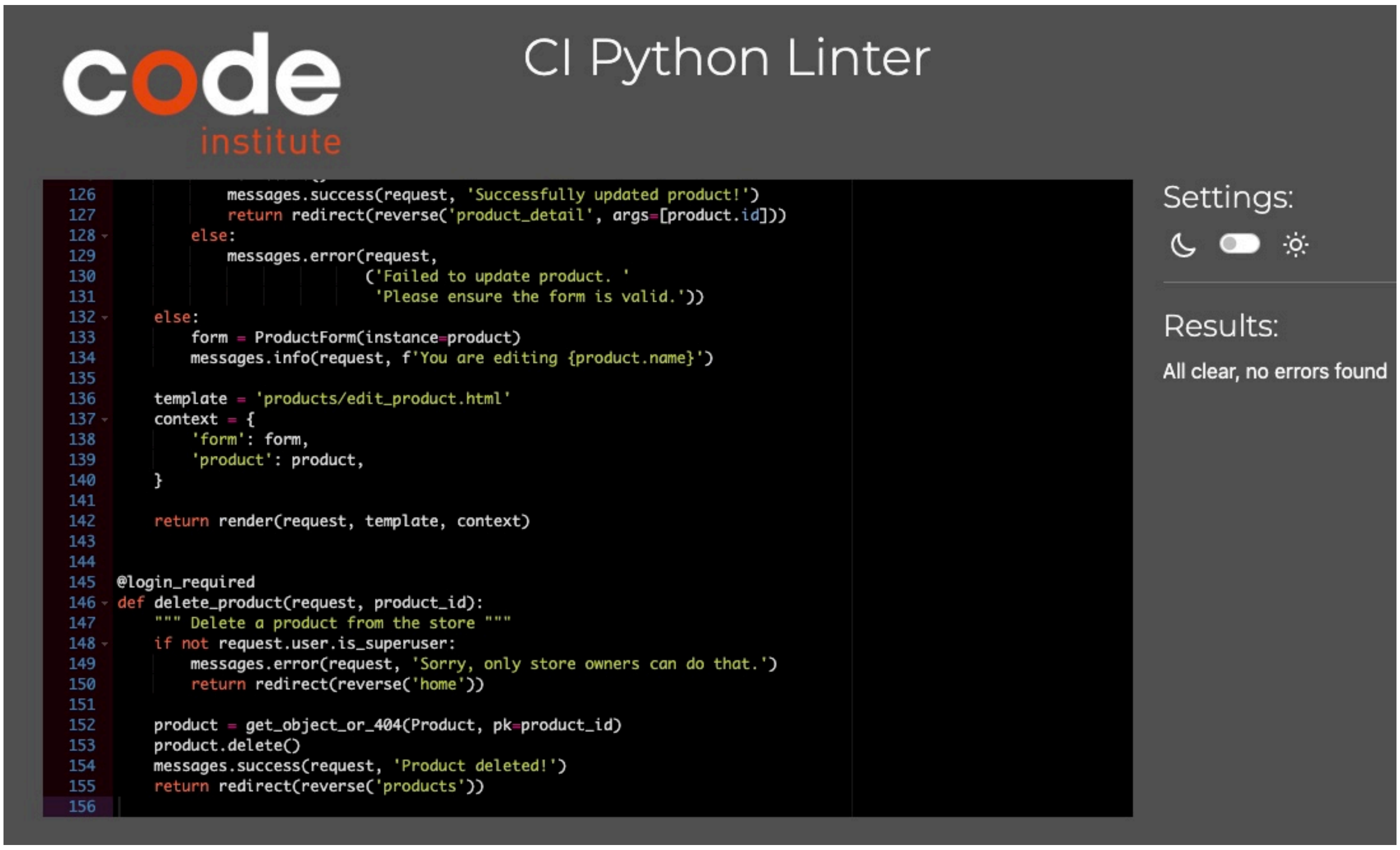
```
        )
    else:
        comment_form = CommentForm(instance=event)
        messages.info(
            request,
            'You are leaving a comment!'
        )


@login_required
def delete_comment(request, comment_id):
    """ Delete a comment, only apply to a superuser """
    if not request.user.is_superuser:
        messages.error(request, 'Sorry, Permission denied.')
        return redirect(reverse('events'))

    # get the comment by the id
    comment = get_object_or_404(Comment, pk=comment_id)

    # get the page user currently at
    event_id = comment.event.id

    # delete the comment
    comment.delete()

    # send user a message
    messages.success(request, f'Comment {comment.body} deleted.')

    # redirect to the page
    return redirect(reverse('event_details', args=[event_id]))
```

Settings:

Results:

All clear, no errors found

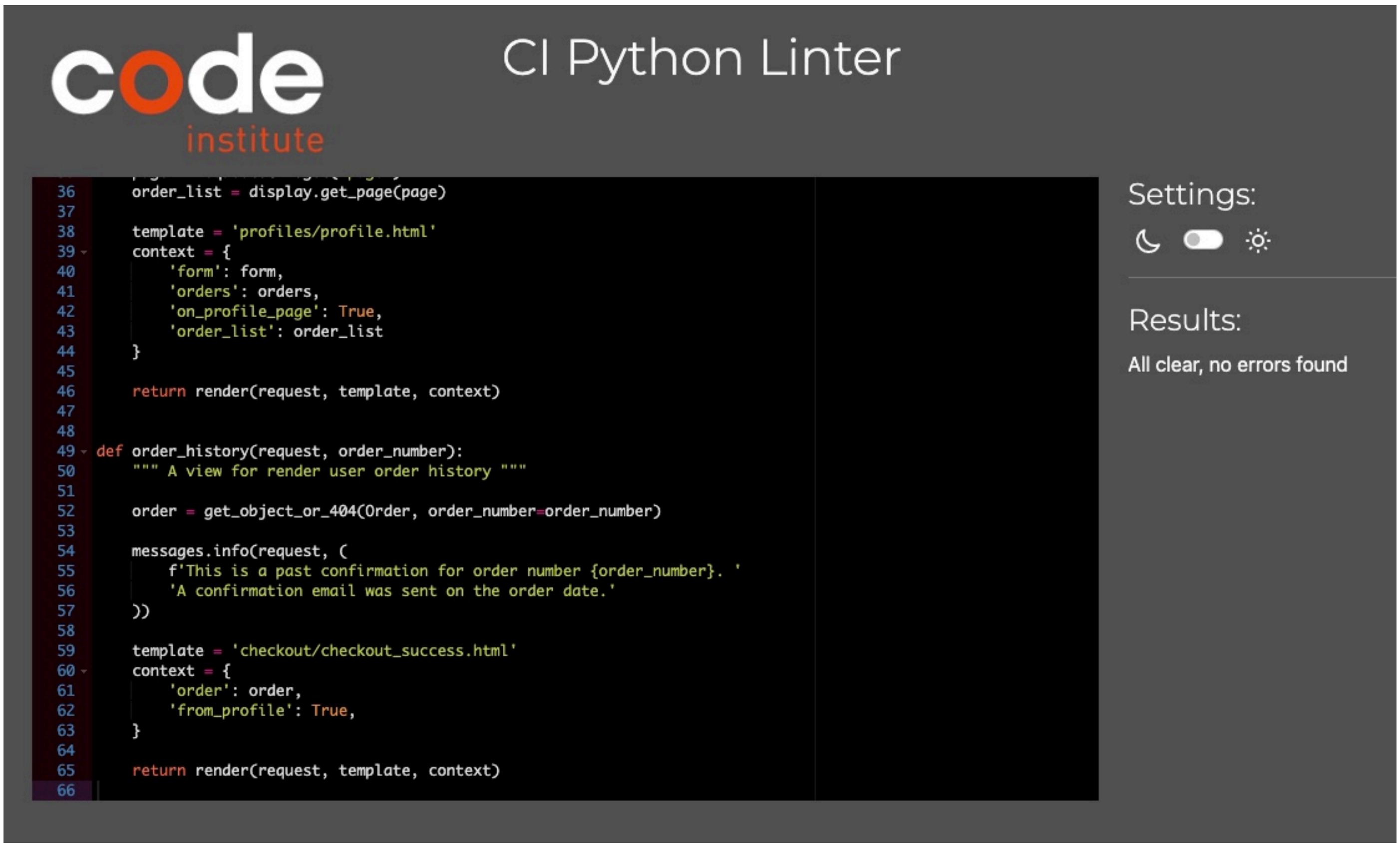
Product app



```
            messages.success(request, 'Successfully updated product!')
            return redirect(reverse('product_detail', args=[product.id]))
        else:
            messages.error(request,
                           ('Failed to update product. '
                            'Please ensure the form is valid.'))
    else:
        form = ProductForm(instance=product)
        messages.info(request, f'You are editing {product.name}')

    template = 'products/edit_product.html'
    context = {
        'form': form,
        'product': product,
    }

    return render(request, template, context)


@login_required
def delete_product(request, product_id):
    """ Delete a product from the store """
    if not request.user.is_superuser:
        messages.error(request, 'Sorry, only store owners can do that.')
        return redirect(reverse('home'))

    product = get_object_or_404(Product, pk=product_id)
    product.delete()
    messages.success(request, 'Product deleted!')
    return redirect(reverse('products'))
```

All clear, no errors found

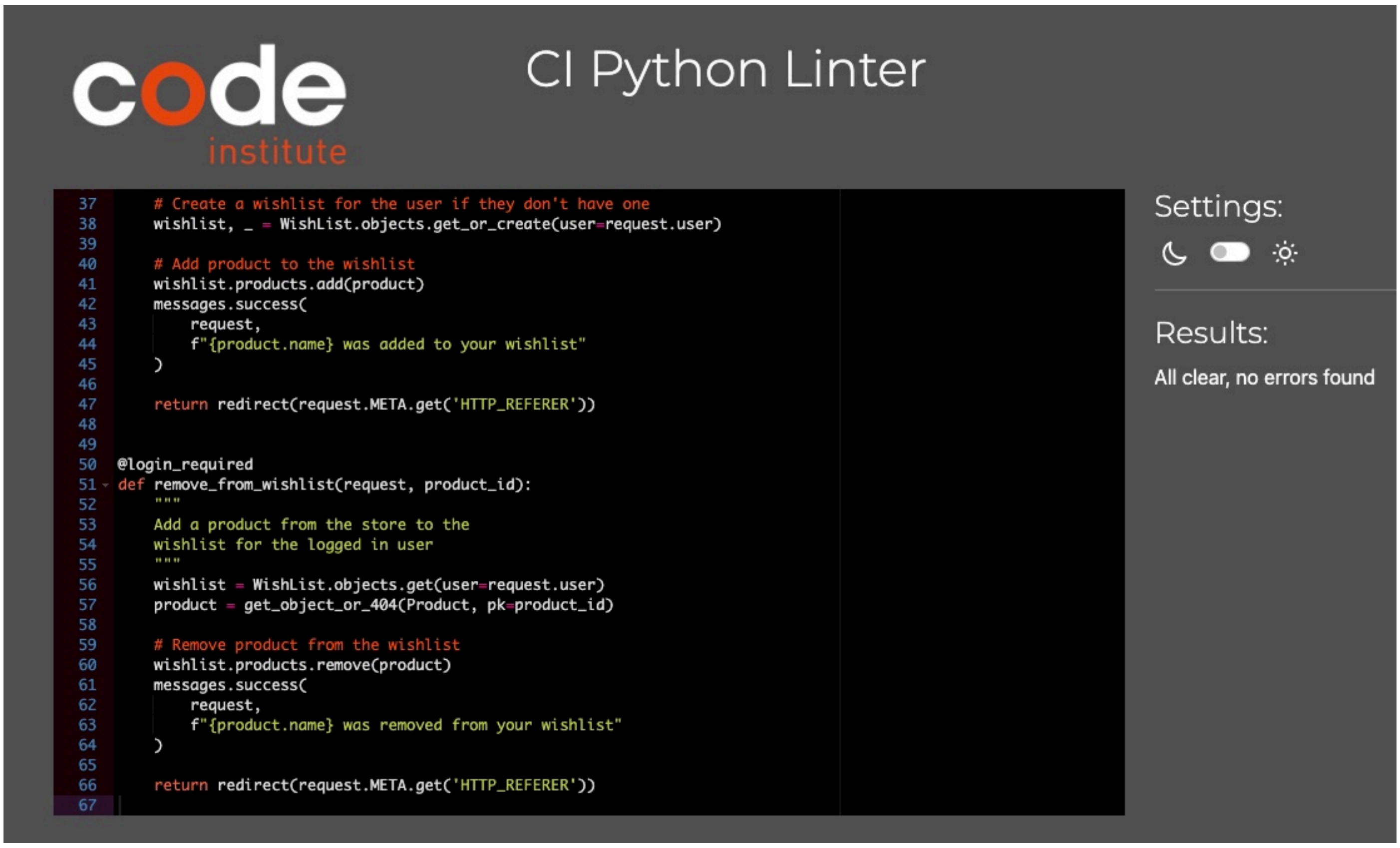
Profile app



```python
    order_list = display.get_page(page)

    template = 'profiles/profile.html'
    context = {
        'form': form,
        'orders': orders,
        'on_profile_page': True,
        'order_list': order_list
    }

    return render(request, template, context)


def order_history(request, order_number):
    """ A view for render user order history """

    order = get_object_or_404(Order, order_number=order_number)

    messages.info(request, (
        f'This is a past confirmation for order number {order_number}. '
        'A confirmation email was sent on the order date.'
    ))

    template = 'checkout/checkout_success.html'
    context = {
        'order': order,
        'from_profile': True,
    }

    return render(request, template, context)
```

Results:
All clear, no errors found

WishList



```
    # Create a wishlist for the user if they don't have one
    wishlist, _ = WishList.objects.get_or_create(user=request.user)

    # Add product to the wishlist
    wishlist.products.add(product)
    messages.success(
        request,
        f"{product.name} was added to your wishlist"
    )

    return redirect(request.META.get('HTTP_REFERER'))


@login_required
def remove_from_wishlist(request, product_id):
    """
    Add a product from the store to the
    wishlist for the logged in user
    """
    wishlist = WishList.objects.get(user=request.user)
    product = get_object_or_404(Product, pk=product_id)

    # Remove product from the wishlist
    wishlist.products.remove(product)
    messages.success(
        request,
        f"{product.name} was removed from your wishlist"
    )

    return redirect(request.META.get('HTTP_REFERER'))
```

Results:

All clear, no errors found