Answers - Create & Update View

Let's now see how to create and update answers from front-end.

Forms

We will create forms to handle our create and update views, the same forms will be used to pass to our views and get data from POST requests.

```python
# Create Answer
class AnswerCreationForm(forms.ModelForm):
    class Meta:
        model = Answer
        fields = ['answer', 'status']


# Answer Update
class AnswerUpdateForm(forms.ModelForm):
    class Meta:
        model = Answer
        fields = ['answer', 'status']
```

Create View

Import the forms into the views file.

```python
# Answer Create
@login_required
def answer_create(request, pk):
    # Check for a valid question
    question = get_object_or_404(Question, pk=pk)
    related_questions = question.tags.similar_objects()[:5]
    if request.method == 'POST':
        # Pass data to form
        form = AnswerCreationForm(request.POST)
        if form.is_valid():
            # Do not save the form yet!
            instance = form.save(commit=False)
            # Set instance parameters —
            # Set author of answer
            instance.author = request.user
            # Link the question
            instance.question = question
            # Set variable
            if not instance.question.has_answers and instance.status ==
'published':
                instance.question.has_answers = True
            # Finally save the instance and commit to database
```

```python
            instance.save()
            # Send a success message
            messages.success(request, 'Answer created successfully!')
            # Redirect to the detail view of newly created answer
            return redirect(reverse('answer_detail', args=
(instance.question.pk, instance.pk,)))
    else:
        # Send a blank form
        form = AnswerCreationForm()
    context = {
        'form': form,
        'question': question,
        'related_questions': related_questions
    }
    return render(request, 'djora/answer_create.html', context)
```

URL's & Templates

Update all the URL's and templates (index, question_detail, answer_list, profile).

Update View

Update view is similar to create view.

```python
# Answer Update
@login_required
def answer_update(request, pk, a_pk):
    # Check if the question exists
    question = get_object_or_404(Question, pk=pk)
    related_questions = question.tags.similar_objects()[:5]
    # Check if the answer exists
    answer = get_object_or_404(Answer, pk=a_pk)
    # Check for authorized user
    if answer.author == request.user:
        if request.method == 'POST':
            # Send data and requested answer
            form = AnswerUpdateForm(request.POST, instance=answer)
            if form.is_valid():
                # Save the updated answer
                instance = form.save(commit=False)
                # Update the related question
                if instance.status == 'published':
                    if not instance.question.has_answers:
                        instance.question.has_answers = True
                # Save and commit the updated answer
                instance.save()
                # Send success message
                messages.success(request, 'Answer updated successfully!')
                return redirect(reverse('answer_detail', args=
(instance.question.pk, instance.pk,)))
        else:
```

```
            form = AnswerUpdateForm(instance=answer)
    else:
        # If current user is not author then raise an error
        raise PermissionDenied
    context = {
        'form': form,
        'question': question,
        'related_questions': related_questions
    }
    return render(request, 'djora/answer_update.html', context)
```

URL's & Templates

Update all the URL's and templates (answer_detail, profile)

Update Sidebar

Since we have functionality for writing answers, let's update all the sidebars with related questions and answer button.