

# Django Guru Hunt Contest

Question গুলো খুব ঠান্ডা মাথায় পড়বে। তাড়াছড়া করবে না। প্রতিটা question মডিউলের কনসেপ্ট এর উপর করা হয়েছে। ঠান্ডা মাথায় question পড়ে উত্তর করার চেষ্টা করবে। শুভকামনা তোমার জন্যে।

mohiuddin.nstu@gmail.com [Switch account](#)



Draft saved

\* Indicates required question

Email \*



Record mohiuddin.nstu@gmail.com as the email to be included with my response

Welcome to Django Guru Hunt Contest



Please Write down your Name in English \*

Mohiuddin Khaled



Please Enter your Course E-mail: \*

mohiuddin.nstu@gmail.com

What is WSGI? \*

1 point

- ☒ a server
- ☐ an interface specifications
- ☐ a Python module
- ☐ a framework

Suppose you want to count the number of books in Django. \*

1 point

```
books = Book.objects.all()
```

Which implementation would be fastest?

- ☒ Database level Implementation – `books.count()`
- ☐ Python Implementation – `len(books)`
- ☐ Template Language Implementation – `{{ books | length }}`
- ☐ None of the above



Which of these variable is the settings for django.contrib.staticfiles app? \*

1 point

- ☐ MEDIA\_URLS
- ☐ MEDIA\_ROOT
- ☒ STATIC\_URL
- ☐ STATIC\_URLS

Which of these is not a way in Form Validation? \*

1 point

- ☐ validate() method
- ☐ clear()
- ☒ clean()
- ☐ None of the above

When we execute form.clean() on a form object and suppose at any point validation error is raised by a validator. What happens after the Validation Error occurs? \*

1 point

- ☐ The program stops executing and Django server stops.
- ☒ The clean() method stops validation and returns a list of validation errors
- ☐ The clean() method continues execution and catches all the validation errors in that form.
- ☐ The clean() method catches the validation error and corrects it



Which command is used to create database migrations in Django? \*

1 point

- ☐ migrate
- ☒ makemigrations
- ☐ sql\_migrations
- ☐ showmigration

What function is used to generate cookies via Django? \*

1 point

- ☐ request.session.get\_cookie()
- ☒ request.session.set\_cookie()
- ☐ request.session.put\_cookie()
- ☐ All of the above

What are the functions available for accessing cookies? \*

1 point

- ☐ request.get\_cookie()
- ☐ request.session\_get\_cookie()
- ☒ request.get.COOKIES()
- ☐ request.COOKIES



What is the difference between media and static files settings? \*

1 point

- ☐ The media settings holds videos and images while static files hold CSS, js.
- ☒ The media setting manages files uploaded by the user. Static settings manages the static assets of the website.
- ☐ Both of the Above
- ☐ None of the Above

What are request.GET and request.POST objects? \*

1 point

- ☐ Python Dictionary
- ☐ Python Object
- ☒ Python Dictionary-Like object
- ☐ Python Object-Like dictionary

To use a Django variable named "username" in a Django template, we use - \* 1 point  
syntax:

- ☐ "username"
- ☒ {{ username }}
- ☐ {% username %}
- ☐ % username %



For a model to be visible in the admin interface, it must be registered in which file?

\* 1 point

- ☐ settings.py
- ☒ admin.py
- ☐ views.py
- ☐ models.py

Given the following Django model: Choose the correct way to filter active users:

\* 1 point

```
class User(models.Model):  
    email = models.EmailField()  
    is_active = models.BooleanField(default=True)
```

- ☐ User.qs.filter(is\_active=True)
- ☒ User.objects.filter(is\_active=True)
- ☐ User.objects().filter(is\_active=True)
- ☐ User.all().filter(is\_active=True)

Which code will give us a text area form field? \*

1 point

- ☐ Field\_name = forms.TextAreaField()
- ☐ Field\_name = forms.CharField(widgets = forms.Textarea)
- ☒ Both of the above
- ☐ None of the above



What is the typical order of an HTTP request/response cycle in Django? \*

1 point

- ☐ URL > view > template
- ☐ form > model > view
- ☐ template > view > model
- ☐ URL > template > view > model

How should we comment crispy forms so that it can work? \*

1 point

- ☐ {% comment %} {{ form | crispy }} {% endcomment %}
- ☐ <!-- {{ form | crispy }} -->
- ☐ # {{ form | crispy }}
- ☐ // {{ form | crispy }}

Render function takes at most \_\_\_\_ parameters. \*

1 point

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4



You're building a blog application with posts and comments. To avoid data duplication and ensure efficient querying, which model structure is best? \* 1 point

- ☐ Create separate models for Post and Comment, with a foreign key in Comment referencing the Post it belongs to.e
- ☐ Store everything in a single model with fields for post content, author, comments, and timestamps.
- ☐ Utilize inheritance, with Post as the base model and Comment inheriting from it with additional comment-specific fields.
- ☐ Implement a nested data structure within the Post model to hold comment data directly.

A user adds an item to their cart in your e-commerce app but leaves without checking out. To encourage completion, which reminder system is optimal? \* 1 point

- ☐ Send an immediate email notification reminding them of the abandoned cart.
- ☐ Implement a pop-up message on their next visit prompting them to resume checkout.
- ☐ Display a personalized banner with the saved items on subsequent website visits
- ☐ All of the above, combined with a timed discount offer for the abandoned cart





For a user-generated content website, which moderation system balances freedom and filtering?(think deeply but be simple) \* 1 point

- ☐ Implement keyword blacklisting and automatic flagging for flagged words.
- ☐ Employ human moderators to review all content before publishing.
- ☐ Utilize a community-based system where users can flag inappropriate content for moderator review.
- ☐ Require strict registration and content verification processes before allowing user submissions.

Scenario: You're designing a social media app with models for Users, Posts, and Comments. How would you represent the relationship between Users and Posts to ensure each Post belongs to a specific User? \* 1 point

- ☐ Create a foreign key field in the Post model, referencing the User model
- ☐ Use inheritance, making Post a subclass of User
- ☐ Store the User's ID as a text field within the Post model.
- ☐ Implement a ManyToManyField between Users and Posts.

Which Django admin method is used to display fields in the admin interface? \* 1 point

- ☐ display\_fields
- ☐ list\_display
- ☐ show\_fields
- ☐ admin\_display



Designing a multilingual news website with categories and tags, what model and URL pattern approach works best?

\* 1 point

- ☐ Create separate models for News, Category, and Tag, with relationships established between them
- ☐ Extend the News model with fields for categories and tags as strings.
- ☐ Implement nested models within News to hold category and tag data
- ☐ Use dynamic URL patterns based on categories and tags for flexible content routing

What is the purpose of Django's manage.py file? \*

1 point

- ☐ a) Manages the database
- ☐ b) Manages static files
- ☐ c) Interacts with Django project via command line
- ☐ d) Handles user authentication

What does the Django HttpResponseRedirect function do? \*

1 point

- ☐ a) Renders a template
- ☐ b) Sends an HTTP response
- ☐ c) Redirects to a specified URL
- ☐ d) Handles form submissions



Scenario: You want to restrict access to a view to logged-in users only. What's the most concise way to achieve this using a decorator?

\* 1 point

- ☐ Use Django's `@login_required` decorator
- ☐ Write a custom decorator that checks for a valid user session
- ☐ Implement manual authentication checks within the view function.
- ☐ Redirect unauthenticated users to the login page using an if statement.

Scenario: You're serving static files (CSS, JavaScript, images) from your Django app. How would you configure your settings to ensure efficient delivery?

\* 1 point

- ☐ Set `DEBUG = True` to serve static files directly from the project directory.
- ☐ Use Django's `staticfiles` app and configure a static files directory
- ☐ Rely on a separate web server like Apache or Nginx to handle static files.
- ☐ Embed static files directly within HTML templates using base64 encoding.

Scenario: You want to display a list of Posts on a template, including the author's name and profile picture for each Post. How would you pass this data from the view to the template?

\* 1 point

- ☐ Use Django's `render()` function to pass a context dictionary containing the data.
- ☐ Access the data directly from the models within the template using model methods.
- ☐ Store the data in session variables and retrieve them in the template.
- ☐ Hardcode the data directly into the template's HTML



Suppose, A teacher can take multiple courses but a course can have only one teacher in a school management system. How do you define this relationship in a Django model? \* 1 point

- ☐ class Course: teacher = models.oneToMany(Teacher)
- ☐ class Teacher: course = models.oneToMany(Course)
- ☐ class Course: teacher = models.ForeignKey(Teacher)
- ☐ class Teacher: course = models.ForeignKey(Course)

Scenario: You're concerned about CSRF attacks on your forms. How would you protect your app against them? \* 1 point

- ☐ Enable Django's built-in CSRF protection and use template tags for secure form rendering
- ☐ Manually validate and sanitize user input before processing it in views
- ☐ Implement custom security filters on forms to block malicious elements.
- ☐ Rely on server-side validation provided by JavaScript libraries.

Submit

Page 1 of 1

Clear form

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#).

Google Forms



