Q1. What are the features of Flask Python?
Flask Python is one of the newest frameworks of Python and is used for designing web applications for the following features:
Flask comes with built-in development server as well as fast debugger
It also contains the integrated support required for unit testing
It has the feature of restful request dispatching
Comes with Jinja2 templating technique
Flask supports secure cookies i.e. client-side sessions
Also has the WSGI 1.0 compliant feature.
It is based on Unicode.
Python Flask is extensively documented.

Q2. What are the advantages of Flask Python?
Flask Python comes with all the advantages of Python and some additional pros of it are:
Flasks design is lightweight and modular. Therefore, it is easy to transform it into the web applications or framework when one needs very few extensions without weighing much.
Flask is ORM-agnostic: i.e. user can plug in their favorite ORM like SqlAlchemy
The basic foundation of API is very nicely shaped and made coherent.
It is very easy to deploy Flask in production as Flask comes with 100% WSGI 1.0 compliant
Flask can handle HTTP request easily with help of its functionalities
It is highly flexible. Its configuration is even more flexible than that of Django, which gives its users plenty of solutions for every product they need.

Q3. What is Flask-WTF and what are their features?
Flask-WTF is featured to offer simple integration with WTForms.
The Features include for Flask WTF are:
Provides Integration with web forms
Is very secure form as it comes with CSRF token
Provides global CSRF protection
Comes with internationalization integration
Also features Recaptcha supporting
Contains File upload that closely works with Flask Uploads

Q4. How long can an identifier be in Flask Python?
In Flask Python, an identifier can be of any length. Also, there are certain rules that the users must follow to name an identifier
It should begin with a character or an underscore or from A-Z or a-z.
The rest of the name of identifier can contain anything from the following: A-Z or a-z or 0-9 or _.
Flask Python is case-sensitive so it will treat upper case and lower case letters differently.
There are certain words, which are reserved in Python called keywords and they cannot be used as

Q5. What are the HTTP methods provided by Python Flask?
HTTP methods are used to retrieve data from an URL:
GET: The GET is the method that sends data to the server unencrypted.
HEAD: HEAD is similar to GET, but that it has no response body.
POST: The POST server does not cache the HTML form data that it sends
PUT: It is the method in which the uploaded content replaces current data representations of the target resources.
DELETE: This method removes the current representations of the target resource that is suggested by a URL.

Q6. What do you mean by template engines in Flask Python?
When users built a website they often face the problem to keep the style of the website consistent. Sometimes the users have had to write multiple times the same text when they ever try to change the style of such websites.
If the website contains only a few pages, changing its style will take the users only some time which is doable.
However, if they have a lot of pages (for example the list of items sold in a mall), this task becomes monotonous and hectic.
Using templates the users may set a basic layout for all their pages and provide which element needs to be changed frequently. Using this way the users can define their header once and keep it consistent in all the pages of their website, and if they need to change their header, they will only have to update it at one place. Making use a template engine will save the users a whole lot of time not only while they create their application but also when they are updating and maintaining it.

Q7. What are the major differences between Pyramid and Flask?
Flask can be stated as a micro framework, which is solely built for a small application, which has simpler requirements. In flask, the users have to use external libraries. Flask is always ready to use.
Pyramid, on the other hand, is built for larger application as it provides flexibility and allows the developer use the right features for their project. The developer can choose the database, templating style URL structure and more. Pyramid is therefore heavy configurable.

Q8. Explain how can one-request database connections in Flask?
Flask framework allows to its users to request database in three ways. They are as follows:
before_request()-These connections are called before making a request and no arguments are passed
after_request() : These connections are called after making a request and response is passed that will be sent to the client.
teardown_request(): These connections are called in a situation where an exception is raised and the response are not sure to get. They are also called after the construction of response. These are not allowed to change the request, and their values can be ignored.

Q9. Explain can a request context be created in Flask Python?
A request profile in Flask Python can be created in two ways:
On its own or auto when the application receives a request from the system
Manually that is by calling app.test_request_context

Q10. Mention how one can enable debugging in Flask Python ?
There are two ways by which users can enable debugging in Flask. They are as follows:
By setting the flag on the applications object
By passing the flag as a parameter to run. If the user is enabling to debug support, the server will reload it when the code will change and the user doesn't have to restart after each change made in the code.

Q11. What do you mean by the Thread-Local object in Flask Python?
Flask Python makes use of thread local objects internally so that the user doesn't have to pass objects around from one function to another function within a request so as to stay thread safe. This approach is quite useful, but it requires a pure request context for dependency injection or while attempting to reuse code, which uses a value indulged in the requests.
Flask Python supports all kinds of database-powered applications like RDBS. Such systems require creating of a schema, which further requires connecting the schema.sql file to a sqlite3 command. So users need to install sqlite3 command if they want to create or start the database in Flask Python.

Q12. How can one-access sessions in Flask Python? State whether Flask Python is an MVC model.

A session in Flask Python is a feature that allows one to remember the information from one request to another. In a flask program, it makes use of a signed cookie so that the user can look at the contents of the session and modify them. The user can also modify the sessions if and only if it has the secret key called the Flask.secret_key. Flask is a small form of Python framework, which behaves the same as the MVC framework. So MVC is a perfect match for Flask.

Q13. What do you mean by a decorator? Name some PDB commands and their uses.

A decorator is defined as a function that adds functionality to another function without changing it. It wraps the function to add some functionality to it.Some PDB commands include
<b>: It adds a breakpoint
<c>: It resumes the execution
<s>: It debugs step by step
<n>: It moves to next line
<l>: It lists the source code
<p>: It prints an expression

Q14. What do you mean by NumPy? Is it better than a list?

NumPy is one of the Flask Python packages which have made its identity in the era of scientific computing. It deals with large data sizes, and also contains a powerful N dimensional array object along with a set of advanced functions.
A NumPy array is much better than a list. Here are the ways:
NumPy is more compact than list.
NumPy is more convenient than the list.
Numpy is more efficient than List.
It is simpler to read and write items with NumPy.

Q15. What do you mean by pickling and unpickling?

To make a portable and serialized representations of Python objects, we have the module known as pickle which accepts a Python object (basically everything in Python is an object) and then converts it into a string type, and after that uses the dump () function to dump it into a file. We term this as pickling.
On the contrary, retrieving objects from the stored string forms is called as unpickling.

Q16. How is memory managed in Flask Python?

Flask Python is a collection of private heap spaces, which holds all objects and data structures together. Programmers cannot access it. It is the task of the interpreter to manage it. But in the core API, users can access some of the tools. The Flask Python memory manager controls its allocation. Also, an inbuilt garbage collector is present which recycles all unused memory so it is made available for the heap space.

Q17. Explain how can you structure a large Flask application ?

To structure a large flask application, one needs to follow these steps:
1. Connect to the functions and then move them to various files, as long as users are assured that it will get imported as the applications start
2. Then use blueprints to assign the views to various categories such as auth, backend, profile, etc.
3. Then use the hidden Werkzeug URL map and then register functions on a central URL map.

Q18. What are the requirements to create the database in Flask Python?

Flask Python supports all kinds of database-powered applications like RDBS. Such systems require creating of a schema, which further requires connecting the schema.sql file to a sqlite3 command. So users need to install sqlite3 command if they want to create or start the database in Flask Python.

19)What is ORM?

Flask SQLAlchemy is an ORM tool which establishes the relationship between the objects and the tables of the relational databases.
The object-relational mapping is the technique of storing python objects into the database tables without writing the raw SQL queries.

20) What is Flask Sijax?

Flask Sijax is a Simple Ajax & jQuery library. It is used to enable Ajax in web applications. It uses JSON to pass data between the server and the browser.

21) How can we get a query string from the Flask?

We can get a query string from the flask by using following function.

```
@app.route('/data')
   def data ( ) :
          user = request.arg.get ('user')
```

22) How can we create request context in Flask?

We can create request context by using following ways.

Automatically when the application receives a request
OR manually, by calling app.test_request_context ('/route?param=value)

23) How can we create structure of large Flask application?

We can create structure of large Flask application by using following steps:

attach to the functions and move them to different files.
Use blueprints to assign the views to "categories". For instance auth, profile, backend, etc.
Use the underlying Werkzeug URL map and register functions on there on a central URL.

24) What are the attributes of request objects?

There are various attributes of request objects:

Attributes	**Description**
Form	It is a dictionary object containing key.
Args	It parsed contents of query string which is part of URL after question mark (?).
Cookies	Dictionary object holds Cookie names and values.
Files	Data pertaining to uploaded file.
Method	Current request method.

25) What are the Mail class methods?

There are following Mail class method:

send(): It is used to send contents of Message class object.
connect(): It is used to opens connection with mail host.
send_message(): It is used to sends message object.

26) What is the default port of Flask?
The default port of Flask is 5000.

27) What is url_for() function in Flask?
In Flask, url_for() function is used to build dynamic URL for specific function.

28) What are the HTTP methods in Flask?
In Flask, the HTTP methods are given below:

GET : It is used to send the data in unencrypted form to the server.
HEAD : It is same as GET, but without response body.
POST: It is used to send HTML from data to server. Data received by POST method.
PUT : It is used to replaces all the current representation uploaded content
DELETE : It is used to removes all current reorientation.

29) What is the default route request in Flask?
In Flask, GET is the default route request.

30) What are the delimiters used in Jinga2 template?
{% ... %}: It is used for Statements
{{ ... }}: It is used for Expressions to print to the template output
{# ... #}: It is used for Comments not included in the template output
... ## : It is used for Line Statements

31) What is the use redirect() function.
Redirect() function is used to display the login page again when a login attempt fails.

32). What is the common way for the Flask script to work?
The common way for the flask script to work is
Either it should be the import path for your application
Or the path to a Python file

33). How can you secure your SQL statement in Flask from SQL injection?
Ans: You can use questions marks while building SQL statements, it will protect SQL statements from the vulnerability of SQL injections.

34). In Flask, explain how does the view function will pass the entries?
Ans: In Flask, the view function will pass the entries as dicts to the show_entries.html template and return the rendered one.

35). Explain how do you get a query string from the flask?
Ans: To get query string from flask, here we want to get the value of user so we will follow the steps as below

```
from flask import request

@app.route(‘/data’)

def data ( ) :

user = request.arg.get (‘user’)
```

36. Explain how you can show all errors in the browser for the Flask?

Ans: To show all errors in the browser for the Flask, you need to run the Python file on the shell. The command used to see errors in detail is “app.debug = True”

37. Explain how can you structure a large Flask application?

Ans: To structure a large flask application, here are the following steps

Stick to the functions and move them to different files, as long as you are assured that it will get imported when the applications starts

Use blueprints to assign the views to “categories”. For instance auth, profile, backend, etc.

Use the underlying Werkzeug URL map and register functions on there on a central URL map

38) What is g in python flask?

g is an object provided by Flask. It is a global namespace for holding any data you want during a single app context. For example, a before_request handler could set g.user, which will be accessible to the route and other functions.

```
from flask import g
@app.before_request
def load_user():
    user = User.query.get(request.session.get("user_id"))
    g.user = user

@app.route("/admin")
def admin():
    if g.user is None or not g.user.is_admin:
        return redirect(url_for("index"))
```

39) How do you enable logging in python flask?

Standard Python logging is used in flask framework for logging. You can log messages about your Flask Application using app.logger, which takes the same name as app.name.

Below is an example to log your own messages in Python Flask

```
def login():
    user = get_user(request.form['username'])

    if user.check_password(request.form['password']):
        login_user(user)
        app.logger.info('%s logged in successfully', user.username)
        return redirect(url_for('index'))
    else:
        app.logger.info('%s failed to log in', user.username)
        abort(401)
```

40)What is the use of jsonify() in Flask?

One of the functions in the flask.json module is jsonify().
It converts data to JSON format and encapsulates it in a Response object with the application/JSON mime-type.
It's worth noting that jsonify is occasionally imported directly from the flask module rather than the flask itself.
In other words, jsonify() is a Flask assist method for appropriately returning JSON data.
jsonify() provides a Response object with the application/JSON mime-type set, whereas json.dumps() only returns a JSON data string.
This could have unforeseen consequences.

41).What is the difference between flask and flask RESTful? How do we create a REST API using a flask?

Flask is a Flask addon that allows us to easily create APIs with a clean interface structure.
It's an agile, quick, and lightweight abstraction that works with existing ORMs and frameworks.

If we understand the basics of Flask, we should find Flask-RESTful to be a simple idea to grasp.
To create a REST API, we have to do the following steps:

Import the modules and start up the program
Creating the REST API endpoints
Write methods to read and write data in the CSV file
Test the endpoints using Postman

42).Describe Flask Error Handlers.

In the Flask application, when an error occurs, an HTTP status code will be returned.

If the HTTP status code is between 400-499, that means the error occurs on the client-side data request.
On the other hand, if the HTTP status code is between 500-599, then it means the error occurs on the server-side application.

When an error occurs with an HTTP status code, we can show custom error pages to the user with the help of error handlers.
It can return responses when a certain type of error is occurred, which is similar to a function view that returns a response when the request URL is similar.
It passes the object to handle the error, in most cases, it is an HTTPException.

Another thing to note is that the response's HTTP status code is not set to the error handler's code.
When returning a response from the error handler, we must make sure to include that.

43) How to get logged user id in flask?
current_user.get_id() is used to get logged in user id in flask.
Here is an example

```
from flask import g
if current_user.is_authenticated():
        g.user=current_user.get_id()
```

22)What is WSGI?

WSGI stands for web server Gateway Interface. It is a specification that describes how a web server communicates with web applications, and how web applications can be chained together to process one request.