1. App folder b.
2. .gitignore file
3. c. Manage.py file
4. d. Requirements.txt file
5. e. Sql files
6. 4. In our files, the SQL files are irrelevant as they are just used to insert dummy data into the database.

7. Manage.py file is the most important file of the project. It manages how the project will run.

10. We have the following functions –

a. FlaskGroup() that can be used to create commands for managing this application - i.

b. We have functions recreate\_db() and seeder() that resets the DB where the recreate\_db() functions deleted and then recreated the db and seeder() function re-enters the dummy data into the newly created database –

11. The recreate\_db() and seeder() are then used in a function called rsd() which also has a decorator @cli.command which means that this function can be used from the command line / terminal directly with the following command - a. python manage.py rsd

12. Similarly, we have another command to run the application called - a. python manage.py run

13. These commands are created with the FlaskGroup() function from earlier.

14. In the app folder, we have the following –

15. The config.py file is used to set some configuration for our project. It is mostly just defining different configurations for different environments like development, stage or production and also to define the URL of the database or which database, URLs or Ports to use.

16. \_\_init\_\_.py file is the most important file of the project. It initialises the app folder so that it can work as a Python module. 17. At first, there are just some modules imported and some functions initialised –

24. The api and the views folder contain code for routes for APIs and Views just similar to how you have learnt before in FLASK.

25. The static folder contains all the static images and files for the application, including the user uploaded attachments.

26. The templates folder contains the HTML, CSS and JavaScript files for the frontend pages of the application.

27. The models folder contains python files for all the tables, that contains the column names and their types, through which, the schema of each and every table in our database is defined.

28. Explore the code and based on all the vulnerabilities you have exploited before, find out where that piece of code is and how it could have been written differently to avoid getting hacked .