1. Create a category:

- URL: `course/category-create`
- Method: POST
- Description: Creates a new category.

2. View all categories:

- URL: `course/category-list`
- Method: GET
- Description: Retrieves a list of all categories.

3. Publish a category:

- URL: `course/category/publish/<id>`
- Method: POST
- Description: Publishes a specific category identified by `<id>`.

4. Delete a category:

- URL: `course/category/delete/<id>`
- Method: DELETE
- Description: Deletes a specific category identified by `<id>`.

5. View courses by category:

- URL: `course/category-wise/list/<cat_id>`
- Method: GET
- Description: Retrieves a list of courses belonging to a specific category identified by `<cat_id>`.

6. View details of a specific category:

- URL: `course/category/<cat_id>`
- Method: GET
- Description: Retrieves the details of a specific category identified by `<cat_id>`.

7. Publish a course:

- URL: `course/publish/<id>`
- Method: POST

- Description: Publishes a specific course identified by `<id>`.

8. Create a course:

- URL: `course/course-create`
- Method: POST
- Description: Creates a new course.

9. View all courses:

- URL: `course/course-list`
- Method: GET
- Description: Retrieves a list of all courses.

10. View courses of a user:

- URL: `course/user/course-list`
- Method: GET
- Description: Retrieves a list of courses belonging to a specific user.

11. View details of a specific course:

- URL: `course/course-detail/<id>`
- Method: GET
- Description: Retrieves the details of a specific course identified by `<id>`.

12. View details of a specific course (admin):

- URL: `course/admin/course-detail/<id>`
- Method: GET
- Description: Retrieves the details of a specific course identified by `<id>` (admin access).

13. Delete a course:

- URL: `course/course-delete/<id>`
- Method: DELETE
- Description: Deletes a specific course identified by `<id>`.

14. Update/edit a course:

- URL: `course/course-edit/<id>`
- Method: PUT

- Description: Updates/edit a specific course identified by `<id>`.

15. View courses of a teacher:

- URL: `course/teacher/course-list/<id>`
- Method: GET
- Description: Retrieves a list of courses belonging to a specific teacher identified by `<id>`.

16. Create a chapter:

- URL: `course/chapter-add`
- Method: POST
- Description: Creates a new chapter.

17. Delete a chapter:

- URL: `course/chapter-delete/<id>`
- Method: DELETE
- Description: Deletes a specific chapter identified by `<id>`.

18. View all chapters of a course:

- URL: `course/<int:course_id>/chapter-list/`
- Method: GET
- Description: Retrieves a list of all chapters belonging to a specific course identified by `<course_id>`.

19. View all chapters of a course (admin):

- URL: `course/admin/<int:course_id>/chapter-list/`
- Method: GET
- Description: Retrieves a list of all chapters belonging to a specific course identified by `<course_id>` (admin access).

20. View details of a specific chapter:

- URL: `course/chapter/<chapter_id>`
- Method: GET
- Description: Retrieves the details of a specific chapter identified by `<chapter_id>`.

21. Complete a chapter video:

- URL: `course/chapters/<int:chapter_id>/complete/`
- Method: POST
- Description: Marks a specific chapter video as complete.

22. Enrol in a course:

- URL: `course/enroll/<course_id>`
- Method: POST
- Description: Enrolls the user in a specific course identified by `<course_id>`.

23. Check enrollment status of a course for a user:

- URL: `course/enroll/check/<course_id>`
- Method: GET
- Description: Checks the enrollment status of a specific course identified by `<course_id>` for the user.

24. Unenroll from a course:

- URL: `course/unenroll/<course_id>`
- Method: POST
- Description: Unenrolls the user from a specific course identified by `<course_id>`.

25. View enrolled courses:

- URL: `course/enrollment/<course_id>`
- Method: GET
- Description: Retrieves the list of enrolled courses for the user.

26. View courses enrolled by a user:

- URL: `course/enrollment/user/<user_id>`
- Method: GET
- Description: Retrieves the list of courses enrolled by a specific user identified by `<user_id>`.

27. View students enrolled in a course:

- URL: `course/enrollment/course/<course_id>`
- Method: GET
- Description: Retrieves the list of students enrolled in a specific course identified by `<course_id>`.

28. Add a course to the wishlist:

- URL: `course/wishlist/`
- Method: POST
- Description: Adds a course to the user's wishlist.

29. Get the wishlist for a specific course:

- URL: `course/wishlist/get/<course_id>`
- Method: GET
- Description: Retrieves the wishlist status for a specific course identified by `<course_id>`.

30. Remove a course from the wishlist:

- URL: `course/wishlist/remove/<id>`
- Method: DELETE
- Description: Removes a specific course from the user's wishlist identified by `<id>`.

31. View all courses in the wishlist:

- URL: `course/wishlist-list-all/`
- Method: GET
- Description: Retrieves a list of all courses in the user's wishlist.

32. Add a review for a course:

- URL: `course/<course_id>/reviews/`
- Method: POST
- Description: Adds a review for a specific course identified by `<course_id>`.

33. View courses by category:

- URL: `course/category-view/<cat_id>`
- Method: GET
- Description: Retrieves a list of courses belonging to a specific category identified by `<cat_id>`.

34. Search for courses:

- URL: `course/search/<query>`
- Method: GET

- Description: Searches for courses based on a specific query.

35. Admin Login:

- URL: `/admin/login/`
- Method: POST
- Description: Performs login for the admin user.

36. Retrieve all users:

- URL: `/get/users/`
- Method: GET
- Description: Retrieves a list of all users.

37. Get information of the authenticated user:

- URL: `/user/`
- Method: GET
- Description: Retrieves the information of the authenticated user.

38. View details of a specific user:

- URL: `/view/user/<user_id>`
- Method: GET
- Description: Retrieves the details of a specific user identified by `<user_id>`.

39. Delete a user:

- URL: `/user/delete/<id>`
- Method: DELETE
- Description: Deletes a specific user identified by `<id>`.

40. Block a user:

- URL: `/user/block/<id>`
- Method: POST
- Description: Blocks a specific user identified by `<id>`.

41. Edit user information:

- URL: `/user/edit/<id>`
- Method: PUT
- Description: Edits the information of a specific user identified by `<id>`.

42. User Login:

- URL: `/auth/user/login/`
- Method: POST
- Description: Performs login for a user.

43. Register as a teacher:

- URL: `/teacher/register`
- Method: POST
- Description: Registers as a teacher.

44. Verify teacher registration request:

- URL: `/request/teacher/verify/<id>`
- Method: POST
- Description: Verifies the teacher registration request identified by `<id>`.

45. Reject teacher registration request:

- URL: `/request/teacher/reject/<id>`
- Method: POST
- Description: Rejects the teacher registration request identified by `<id>`.

46. Retrieve all teachers:

- URL: `/teacher/get`
- Method: GET
- Description: Retrieves a list of all teachers.

47. Retrieve details of a specific teacher:

- URL: `/teacher/get/<id>`
- Method: GET
- Description: Retrieves the details of a specific teacher identified by `<id>`.

48. View teacher registration request:

- URL: `/request/teacher/<id>`
- Method: GET
- Description: Retrieves the teacher registration request identified by `<id>`.

49. Submit teacher registration request:

- URL: `/request/teacher/`
- Method: POST
- Description: Submits a teacher registration request.

50. Live View:

- URL: `live/`
- Method: GET
- Description: Retrieves the live view.

51. Delete Live:

- URL:`live/<room_code>`
- Method: DELETE
- Description: Deletes a specific live identifier by `<room_code>`.

52. Quiz View:

- URL:`course/quiz/`
- Method: GET
- Description: Retrieves the quiz view.

53. Quiz Attempt:

- URL: `course/quiz/attempt/<course_id>`
- Method: POST
- Description: Attempts a quiz for a specific course identified by `<course_id>`.

54. Create a Banner:

- URL: `/banner/`
- Method: POST
- Description: Creates a new banner.

55. View Banners for Users:

- URL: `/banner/user`
- Method: GET
- Description: Retrieves a list of banners for users.

56. Edit a Banner:

- URL: `/banner/<id>`

- Method: PUT
- Description: Edits a specific banner identified by `<id>`.

57. Chatting Endpoint:

- URL: `ws://localhost:8000/ws/chat/<room_id>/`
- Description: Establishes a WebSocket connection for the chat room with the specified `<room_id>`.