To run the Django application locally with the code mentioned above, follow these steps:

1. **Set Up Your Development Environment**

- Ensure you have Python installed (version 3.9 or later).
- Install Docker and Docker Compose

    - **Docker**: Follow the official Docker installation guide for your operating system.
    - **Docker Compose**: Follow the official Docker Compose installation guide for your operating system.

2. **Prepare the Project**

I. Clone the Project Repository-> If you haven't already, clone the project repository:-
git clone <repository-url>
cd task_management_api

II. Create and Activate a Virtual Environment-> It's a good practice to use a virtual environment for Python projects:
python -m venv venv
source venv/bin/activate # On Windows use `venv\Scripts\activate`

III. Install Dependencies-> Install the required Python packages:
pip install django djangorestframework mysqlclient djangorestframework-simplejwt

IV. Create a `.env` File
Create a `.env` file in the root directory of your project with the following content (replace placeholders with actual values):
SECRET_KEY=your_secret_key
DEBUG=True
ALLOWED_HOSTS=localhost, 127.0.0.1
DATABASE_NAME=task_management
DATABASE_USER=user
DATABASE_PASSWORD=password
DATABASE_HOST=localhost
DATABASE_PORT=3306

3. **Run the Application Locally**

Option 1: Using Docker

A. Build Docker Images and Start Containers->Build the Docker images and start the containers:
docker-compose up –build
This command will build the Docker image for your Django application and start the MySQL database container as well.

B. Access the Application->Open your web browser and go to `http://localhost:8000/`. The Django application should be running.

C. Create and Apply Migrations->In a separate terminal, you need to run the following commands to create and apply database migrations:
docker-compose exec web python manage.py makemigrations
docker-compose exec web python manage.py migrate

Option 2: Without Docker

A. Run Migrations->First, ensure the MySQL server is running and configured correctly on your local machine. Use a database client or command line tool to create a database with the name specified in `.env` (`task_management`). Then, run the migrations to set up your database schema: python manage.py makemigrations
python manage.py migrate
B. Run the Django Development Server->Start the Django development server: python manage.py runserver
C. Access the Application->Open your web browser and go to `http://127.0.0.1:8000/`. The Django application should be running.

4. Create Superuser->To access the Django admin panel and manage the application, you need to create a superuser: python manage.py createsuperuser
   Follow the prompts to create the superuser account.
5. Testing the Application->You can use Django's built-in testing framework to run tests: python manage.py test
6. Summary->
   A. **Docker**: Build and start containers using `docker-compose up --build`.
   B. **Local Environment**: Set up the database, run migrations, and start the development server using `python manage.py runserver`.

This should get your Django application up and running locally with the configurations provided.