**1. Setup Your Development Environment - Complete**

* **Django Backend:**
  + Install Django and create a new Django project.
  + Set up a virtual environment for Python dependencies.
* **React Frontend:**
  + Use **create-react-app** to generate a new React application.
  + Configure proxy in **package.json** for local development to forward requests to the Django backend.
* **Docker:**
  + Install Docker Desktop for your operating system.
  + Create a **Dockerfile** for both Django and React.
  + Create a **docker-compose.yml** file to define services, networks, and volumes.

**2. Develop Your Application – Will complete later**

* **Django:**
  + Develop your models, views, and URLs.
  + Configure the Django REST Framework for API development if needed.
* **React:**
  + Develop your components, services, and state management using React.
  + Use Axios or Fetch API to communicate with the Django backend.

**3. Dockerize Your Application – Need to redo – separate front/back end**

* Write Dockerfile for both Django and React:
  + For Django, use the Python base image, copy your project, install dependencies, and run the Django server.
  + For React, use the Node base image, copy your project, install dependencies, and build your React app.
* In **docker-compose.yml**, define services for Django, React, and any other services like databases.

**4. Testing Your Docker Containers**

* Use **docker-compose up** to build and start your services.
* Test the application to ensure both Django and React are properly communicating.

**5. Prepare for Deployment**

* **Heroku:**
  + Sign up for a Heroku account if you haven't already.
  + Install the Heroku CLI.
  + Login to your Heroku account via CLI.
  + Create a new Heroku app.
* **Docker & Heroku:**
  + Configure **heroku.yml** for deploying Docker containers.
  + Push your Docker images to Heroku Container Registry.
  + Use Heroku's PostgreSQL addon for your database or configure another database service.

**6. Deploy to Heroku**

* Deploy your application using Heroku CLI.
* Release the containers to your Heroku application.
* Configure environment variables in Heroku.

**7. Final Checks and Launch**

* Ensure that all components are running correctly on Heroku.
* Perform final testing to check the functionality and performance of your application.
* Update your domain settings if you have a custom domain.

**Conclusion**

This outline provides a broad overview of the steps involved in setting up and launching an application with Django, React, Heroku, and Docker. Each step involves several detailed actions, so it's important to consult official documentation or specific guides for each tool or service you're using.