Tasks (Python + React Native)

1. **Custom Authentication**
   - Implement custom login and registration functionality.
   - Add two new fields to the `users` table:
     - `mobile_no`
     - `is_admin` (boolean flag).
   - Use **Twilio** for mobile number verification during registration. Only verified users can complete registration.
   - Enable **social login** (Google, Facebook, etc.).
2. **Article Management**
   - Both guests and logged-in users can submit articles.
     - Guest submissions will be marked as **unapproved** until admin approval.
   - Admin manages a list of categories.
     - Each article can belong to multiple categories.
     - Categories with related articles cannot be deleted.
   - Admin has full CRUD permissions on articles and categories.
   - Logged-in users can manage **only their own submissions**.
3. **Admin Features**
   - Admin can see a **tabular listing** of all articles.
   - Admin can edit, approve, or delete articles.
   - Protect all admin-specific screens using **custom middleware**.

4. **Frontend (React Native App)**
    - Responsive layout (mobile-first) with **Bootstrap-like styling** adapted for React Native.
    - Articles listing with:
        - Title
        - Excerpt
        - Read more link
        - Number of comments
    - Details page showing:
        - Title
        - Full description
        - Related categories (linked to category-specific listings).
5. **Slug & Permalink**
    - Unique slugs for both **articles** and **categories**.
    - Automatically generate slug on add/edit.
    - Each model should have a permalink attribute.
        - Example:
            - Title: `Hello World` → Slug: `hello-world`
            - Permalink: `/article/hello-world`
            - `http://domain.com/article/hello-world` loads details.
6. **Media & Popularity**
    - Upload images for articles.

    - Manage article popularity.
    - Setting to mark an article for homepage slider.

7. **Pagination**
    - Article listings must be paginated.
    - Items per page controlled via `.env`.
    - Pagination links must navigate correctly to article detail pages.
8. **Comments**
    - Users can submit comments on articles.
    - Guest comments require Twilio validation + admin approval.
    - Approved comments appear below article details.
9. **Notifications**
    - Admin receives email when a new article is submitted.
    - Article authors receive email when a new comment is submitted.
    - Use **Mailtrap** + **Python Markdown email templates**.
10. **Contact Us**
    - "Contact Us" form processed using **AJAX requests** in React Native.

Advanced Features

- **Twilio**:
    - Validate mobile numbers during registration.
    - Guest article/comment submissions require Twilio validation.

- **Stripe Integration**:
    - Users must pay via Stripe before being allowed to submit an article.
- **Template Switching**:
    - Ability to switch between multiple frontend templates/themes for the site.
- **Social Login**:
    - Google, Facebook, etc. integration for sign-in.

Tech Stack

- **Backend (Python)**: FastAPI / Django REST Framework (DRF) with custom authentication, Twilio integration, Stripe integration, Eloquent-like ORM (Django ORM or SQLAlchemy).
- **Frontend (React Native)**: Responsive UI, bootstrap-inspired components, rich text editor for article details.
- **Database**: PostgreSQL / MySQL with migrations.
- **Email**: Mailtrap using Python's email libraries + Markdown.
- **Validation**: Client + Server side validations.
