views.py

These views handle backend logic and return JSON responses. They are typically used by frontend applications or external systems.

`create_case(request)`

- **Purpose:** Creates a new case using data from a JSON POST request.
- **HTTP Method:** POST
- **Input:** JSON payload with the following fields:
  - `patient_name` (string): Name of the patient.
  - `patient_code` (string): Unique code for the patient.
  - `dentist_name` (string): Name of the dentist.
  - `dentistry_type` (string): Type of dentistry.
  - `tooth_numbers` (list of strings): List of tooth numbers.
  - `shade_system` (string): Shade system used.
  - `shade` (string): Shade value.
  - `notes` (string, optional): Additional notes.
- **Output:** JSON response with:
  - `success` (boolean): Indicates if the operation was successful.
  - `message` (string): Success or error message.
  - `case_id` (string, optional): Generated case number (if successful).
    - **Example Request:**
      - {
        "patient_name": "John Doe",
        "patient_code": "12345",
        "dentist_name": "Dr. Smith",
        "dentistry_type": "Crown",
        "tooth_numbers": ["12", "13"],
        "shade_system": "VITA",
        "shade": "A1",
        "notes": "Patient prefers a lighter shade."
        }
    - **Example Response:**
      - {
        "success": true,
        "message": "Case created successfully!",
        "case_id": "CASE12345"
        }
-

`send_to_outsource(request, case_id)`

- **Purpose:** Sends a case to an outsource provider.
- **HTTP Method:** POST
- **Input:**
  - `case_id` (int): ID of the case to be outsourced.
- **Output:** JSON response with:

- `success` (boolean): Indicates if the operation was successful.
- `message` (string): Success or error message.

- **Example Response:**
  - {
    "success": true,
    "message": "Case sent to Outsource successfully!"
    }

2. Page Views

These views render HTML templates for the frontend.

`home(request)`

- **Purpose:** Renders the home page.
- **Template:** `app/index.html`
- **Context Data:**
  - `title`: "Home Page"
  - `year`: Current year.

`contact(request)`

- **Purpose:** Renders the contact page.
- **Template:** `app/contact.html`
- **Context Data:**
  - `title`: "Contact"
  - `message`: "Your contact page."
  - `year`: Current year.

`about(request)`

- **Purpose:** Renders the about page.
- **Template:** `app/about.html`
- **Context Data:**
  - `title`: "About"
  - `message`: "Your application description page."
  - `year`: Current year.

`upload_model(request)`

- **Purpose:** Handles the upload of a 3D model.
- **HTTP Method:** GET (renders form) or POST (processes form).
- **Template:** `upload_model.html`
- **Context Data:**
  - `form`: Instance of `Model3DForm`.

`model_list(request)`

- **Purpose:** Lists all uploaded 3D models.
- **Template:** `model_list.html`
- **Context Data:**
  - `models`: QuerySet of all `Model3D` objects.

`view_model(request, model_id)`

- **Purpose:** Displays details of a specific 3D model.
- **Template:** `model_detail.html`
- **Context Data:**
  - `model`: The `Model3D` object with the given `model_id`.

`cases(request)`

- **Purpose:** Renders the cases management page.
- **Template:** `app/cases.html`
- **Context Data:**
  - `title`: "Cases"

`dashboard(request)`

- **Purpose:** Renders the dashboard page.
- **Template:** `app/dashboard.html`
- **Context Data:**
  - `title`: "Dashboard"

`calls(request)`

- **Purpose:** Renders the calls management page.
- **Template:** `app/calls.html`
- **Context Data:**
  - `title`: "Calls"

`accounting(request)`

- **Purpose:** Renders the accounting management page.
- **Template:** `app/accounting.html`
- **Context Data:**
  - `title`: "Accounting"

`sales(request)`

- **Purpose:** Renders the sales dashboard.
- **Template:** `app/sales.html`
- **Context Data:**
  - `title`: "Sales"
  - `this_month`: "$0.00"
  - `this_year`: "$0.00"
  - `lifetime`: "$0.00"

`designer_dashboard(request)`

- **Purpose:** Renders the designer dashboard.
- **Template:** `app/designer_dashboard.html`
- **Context Data:**
  - `title`: "Designer Dashboard"

`lab_product(request)`

- **Purpose:** Renders the lab product management page.
- **Template:** `app/lab_product.html`

- **Context Data:**
  - `title`: "Lab Product"
  - `tabs`: List of tabs for navigation.

`outsource(request)`

- **Purpose:** Renders the outsource management page.
- **Template:** `app/outsource.html`
- **Context Data:**
  - `title`: "Outsource"

`settings(request)`

- **Purpose:** Renders the settings page.
- **Template:** `app/settings.html`
- **Context Data:**
  - `title`: "Settings"

`accounts(request)`

- **Purpose:** Renders the accounts management page.
- **Template:** `app/accounts.html`
- **Context Data:**
  - `title`: "Accounts"

`search_cases(request)`

- **Purpose:** Handles search functionality for cases and renders the results.
- **Template:** `cases/search_results.html`
- **Context Data:**
  - `results`: Paginated list of matching cases.
  - `query`: The search query.

How to Use This Documentation

1. **For Developers:**
   - Use the documentation to understand the purpose and functionality of each view.
   - Test the API endpoints using tools like Postman or cURL.
   - Ensure the corresponding templates exist in the `templates` directory.
2. **For Frontend Developers:**
   - Use the context data to populate the templates.
   - Call the API endpoints to fetch or submit data.
3. **For API Consumers:**
   - Refer to the API documentation to understand the expected input and output formats.
   - Use the example requests and responses as a guide.