Module Documentation:

Homepage1

It is the home page module for the website.It includes navigation,a caurousel,about us section,feature cards and a footer.

Features

1) Navigation Bar

- Provides links to **Home, About, Contact, Login,** and **Signup** modules.

2) Carousel

- Highlights key aspects of the platform with rotating banners.

3) About Section

- Highlights the purpose and features of Animimic, such as **community forums, blogs, and 3D simulations**.

4) **Feature Cards**

- Showcases sections for **Community**, **Animations**, and **Blogs**, each linking to their respective exploration pages.

5) Footer

- Links to **FAQ, Privacy Policy, Terms**, and **Social Media platforms.**

Homepage2

The HomePage2 component serves as the homepage for logged-in users of Animimic.

Features

1) **Navigation Bar**

- Links to **Home**, **About**, **Contact**, and **Profilepage** modules.

2) **Carousel**

- Highlights key aspects of the platform with rotating banners.

3) **About Section**

- Describes the purpose and unique features of Animimic, including **community forums, blogs, and 3D pet simulations.**

4) **Feature Cards**

- Showcases sections for **Community**, **Animations**, and **Blogs**, each linking to their respective exploration pages.

5) **Footer**

- Provides **FAQs, contact information, privacy policies,** and links to **social media platforms.**

Login

The Login Component in the application serves as the authentication interface, allowing users to securely access their accounts by providing their credentials (**username, email, and password**).

Features

1. **User Authentication**
   - Validates and submits login credentials to the backend API.
2. **Input Validation**
   - Ensures all required fields (username, email, password) are filled.
3. **Navigation**
   - Redirects to the **Homepage2** after successful login.
4. **Module Links**
   - Provides links to **Register** and **Reset Password** modules.

Register

The **Register** module serves as the user registration interface for the application. Its purpose is to collect necessary user information (like **name, username, mobile, email, password**) and validate it.

Features

1) **User Registration**

- Collects essential details such as **name, username, mobile, email, and password**.

2) **Form Validation**
   - Ensures all required fields are filled and validates the format of the email and mobile number.
3) **Backend Integration**
   - Submits the form data to the backend API and handles success/error responses.
4) **Module Links**
   - Provide links to **terms and conditions** page to read ,to accept the terms and conditions.
5) **Navigation**
   - Redirects the user to the **Security Questions** page upon successful registration, or allows them to navigate to the **Login** page if they already have an account.

Profile Management

The Profile Management module allows users to manage their account settings after logging in.

Features

1) **Change Password**
   - Users can update their password with a form, with gets redirected to **Reset-password** module.
2) **Delete Account**
   - A button for users to delete their account .After deletion redirects to **Homepage1** module.
3) **Logout**
   - Users can log out and be redirected to the **Homepage1** module.

SecurityQuestions

The SecurityQuestions module allows users to set security questions and answers as an additional layer of account protection. After login, the user is prompted to fill in three security questions, which are then saved to the backend for future account recovery or verification.

Features

1) **Security Question Selection**
   - Users can **select** questions from a predefined list.

2) **Answer Input**
   - Users input their corresponding answers for each question.
3) **Validation**
   - Ensures all fields are filled before submission.
4) **Backend Integration**
   - Submits the entered data to the backend for storage and updates.
5) **Navigation**
   - After successful submission, the user is redirected to the **Login** page.

TermsAndConditions

The **Terms and Conditions** module provides important legal information regarding the use of the Animimic Platform. It outlines user rights and responsibilities, usage rules, and platform features, ensuring users are informed of their obligations while interacting with the site.

Features

1) **Information**
   - Conatins information related to topics **Introduction,User License and Content,Access and Use, Verification of Account Information,Privacy,Third Party Websites and Promotions,Warranties and Disclaimer,Limitations of Liablity,Indemnification, Termination / Deletion of Accounts,Information , Complaints & Copyright Claims, Disputes,Applicable Law and Competent Court**
2) **Module Links**
   - Contains links to **privacy policy** and **cookie policy modules** .

PrivacyPolicy

This module informs users about how their personal data is collected, used, and shared while using Animimic's platform, which includes reading/uploading blogs, participating in forums, and engaging with 3D simulations.

Features

1) **Module Links**
   - Contains link to **terms and conditions module**.

CookiePolicy

The Cookie module provides users with essential information about how Animimic uses cookies, what data is collected, and how users can manage their cookie preferences to enhance their experience on the platform.

Features

1) **Module Links**
    - Contains link to **Privacy Policy module**.

About Us

The About Us module introduces users to the Animimic platform, highlighting its mission, core values, and features. It appears after login and serves as a welcoming page, offering insights into the community and services.

Features

1) **Core Values Section**
    - Highlights Animimic's emphasis on community, learning, and fun.
2) **Animimic Offers**
    - Describes key platform offerings, including:
3) **Module Links**
    - Contains link to **Login module.**

Frequently asked questions

The FAQ component is designed to provide users with quick answers to common questions about the **Animimic** platform.

Features

1. **Comprehensive FAQs**
    - Covers topics like **account creation, community features, blog uploads, security, and more**.

Reset-Password

The Reset-Password module allows users to securely reset their passwords by providing their email, a new password, a selected security question, and its answer.

Features

1) **Form Validation**
   - Ensures all required fields (**email, password, security question, and answer**) are filled before submission.
2) **Dynamic Interaction**
   - Displays warnings for missing fields using toast notifications.
3) **API Integration**
   - Sends user-provided data to the backend endpoint for password reset.
4) **Error Handling**
   - Handles API errors gracefully, resetting fields and displaying error messages.

5) **Navigation**
   - Successfully resetting the password navigates users to the homepage.

Dog

The Dog module is part of an interactive animal simulation module in a web application. After the user logs in, this page allows them to interact with a 3D dog model, control its actions

Features:

1) **3D Dog Model**
   - Displayed in a Three.js scene with interactive controls for jumping and rotating.
2) **Sound Controls**
   - Users can make dog mimic other animal's sounds.
3) **Text-to-Speech**
   - Users can type text and have it read aloud using the browser's speech synthesis API.
4) **Navigation**
   - The user can navigate to other animal modules (e.g., **Elephant, Cat, Tiger, Bee**) via button.
5) **Responsive Controls**
   - Buttons for jumping, rotating, and playing sounds are easily accessible

Cat

The Cat Module allows users to interact with a 3D animated cat and control its actions.

Features:

1) **3D Cat Model**
    - Displayed in a Three.js scene with interactive controls for jumping and rotating.
2) **Sound Controls**
    - Makes cat mimic other animals sounds
3) **Text-to-Speech**
    - Users can type text and have it read aloud using the browser's speech synthesis API.
4) **Navigation**
    - Links to other animal modules like Dog, Elephant, Tiger, and Bee.
5) **Responsive Controls**
    - Buttons for jumping, rotating, and playing sounds are easily accessible

Elephant

The Elephant Module allows users to interact with a 3D animated elephant and control its actions.

Features:

1) **3D Elephant Model**
    - Displayed in a Three.js scene with interactive controls for jumping and rotating.
2) **Sound Controls**
    - Users can make elephant mimic other animals sounds
3) **Text-to-Speech**
    - Users can type text and have it read aloud using the browser's speech synthesis API.
4) **Navigation**
    - Links to other animal modules like **Cat**, **Elephant, Tiger, and Bee**.
5) **Responsive Controls**
    - Buttons for jumping, rotating, and playing sounds are easily accessible

Bee

The Bee Module allows users to interact with a 3D animated bee and control its actions.

Features

1) **3D Bee Model**
    - Displayed in a Three.js scene with interactive controls for jumping and rotating.
2) **Sound Controls**
    - Users can make bee mimic other animals sounds
3) **Text-to-Speech**
    - Users can type text and have it read aloud using the browser's speech synthesis API.
4) **Navigation**
    - Links to other animal modules like **Cat**, **Elephant, Tiger, and Dog**.
5) **Responsive Controls**
    - Buttons for jumping, rotating, and playing sounds are easily accessible

Tiger

The Tiger Module allows users to interact with a 3D animated bee and control its actions.

Features

1) **3D Tiger Model**
    - Displayed in a Three.js scene with interactive controls for jumping and rotating.
2) **Sound Controls**
    - Users can make Tiger mimic other animals sounds
3) **Text-to-Speech**
    - Users can type text and have it read aloud using the browser's speech synthesis API.
4) **Navigation**
    - Links to other animal modules like **Cat**, **Elephant, Bee, and Dog**.
5) **Responsive Controls**
    - Buttons for jumping, rotating, and playing sounds are easily accessible

Discussion Forum

The Discussion Forum module provides a platform for users to engage in discussions on various topics after logging in. It encourages interaction through posts, comments, and reactions, fostering a community-driven experience.

Features

1) **Create New Discussions:**
    - Users can create new posts with a title, description, and detailed content.
    - Posts are tied to the logged-in user's ID for identification.
2) **Search Functionality:**
    - A search bar enables users to filter discussions based on the description.
3) **Engagement Tools:**
    - Like and dislike functionality for posts with real-time updates.
    - Comment button to navigate to a dedicated comments page for the selected discussion.
4) **Modal for Post Creation:**
    - A user-friendly modal interface for creating new posts.
    - Form validation ensures all required fields are filled before submission.
5) **Dynamic Updates:**
    - Discussions dynamically update when posts are created or reactions are added.
6) **Optimistic UI:**
    - Immediate updates to likes/dislikes before server confirmation for better user experience.

CommentPage

The CommentPage module allows users to view, add, and interact with comments and replies on a discussion post. It enhances user engagement by facilitating threaded discussions within the app.

Features:

1) **Discussion Details**:
    - Displays the title and content of the selected discussion post.
2) **Add Comments**:
    - Users can add new comments to the discussion.
    - Includes a textarea and a button to submit comments.
3) **Threaded Replies**:

- Users can reply to individual comments.
- Replies are nested under their respective comments.

4) **Navigation**
   - Linked from the homepage or feed page where discussions are listed. Clicking on a discussion navigates to this page

.

Navbar

The Navbar module serves as the primary navigation bar for the application, enabling users to access various sections, perform searches, and interact with key features.

Features

1) **Logo and Home Navigation:**
   - Clicking the logo redirects users to the homepage.
2) **Search Functionality:**
   - Allows users to search for content directly from the navbar.
   - Automatically navigates to search results on pressing Enter.
3) **Quick Links:**
   - My Blogs: Access the user's personal blog dashboard(not implemented).
   - Write: Navigate to the editor page for creating new posts.
4) **User Interaction:**
   - User profile button for to navigate to user’s profile.

HomePage

The HomePage module serves as the main landing pagefor blogs page, displaying the latest blogs, trending blogs, and category-based blog filtering options.

Features

1) **Dynamic Blog Display**:

- Shows the **latest blogs and trending blogs** fetched from the backend.

2) **Category-Based Filtering**:
   - Allows users to **filter blogs** by specific categories dynamically.
3) **Load More Functionality**:
   - Supports pagination for fetching and displaying **additional blogs**.
4) **Trending Blogs Section**:
   - Highlights **popular and trending** blogs.
5) **Fallback and Error Handling**:
   - Displays loaders while fetching data and handles cases with **no data** gracefully using messages.

Connections to Other Modules:

- **Blog Post Components**:

  Utilizes **BlogPostCard** and **MinimalBlogPost** to render blog previews.

- **Navigation**:

  Works alongside **InPageNavigation** for transitions between blog types.

SearchPage

The SearchPage module is designed to handle search functionality for blogs based on user queries. It retrieves matching blog results from the server, displays them, and supports pagination for viewing more results.

Features:

- **Search Query Handling:** Fetches blog results based on a query passed through the URL parameter.
- **Blog Display:** Renders blog posts as cards using the BlogPostCard module.
- **Pagination:** Supports loading more results with the LoadMoreDataBtn module.
- **Error Handling:** Displays a loader while data is being fetched and shows a message if no blogs are found.

- **In-page Navigation:** Includes an InPageNavigation module to display the search context and allow easy navigation within the page.

Connections with Other Modules:

- **InPageNavigation:** Provides the navigation header for the search results.
- **BlogPostCard:** Displays each blog post as a card.
- **Loader:** Shows a loading spinner while data is being fetched.
- **LoadMoreDataBtn:** Handles fetching and displaying additional pages of search results.
- **NoDataMessage:** Displays a message when no results are found.
- **filterPaginationData:** Formats and updates the state with new blog data and pagination information.

BlogPostCard

The BlogPostCard component is designed to display a summary of individual blog posts on the homepage. It provides users with a quick preview of key blog details such as title, description, author, tags, and likes, along with a thumbnail image. Clicking on the card navigates the user to the full blog post.

Features

1) **Dynamic Content Rendering:**
   - Displays blog details like the title, author name, username, description, tags, and likes.
   - Handles missing data gracefully with defaults like "Unknown Author" or "Anonymous."
2) **Interactive Elements:**
   - Links to individual blog pages for full content.
3) **Author Info:**
   - Displays the author's name and username
4) **Visual Appeal:**
   - Shows a thumbnail (banner image) for the blog post.

BlogPage

The BlogPage module serves as the detailed view for individual blog posts. It fetches and displays blog content, author details, and other metadata, along with handling user interactions like likes. It also shows similar blog posts based on tags, providing users with related content.

Features:

1) **Blog Content Display**
   - Fetches and displays blog details, including title, content, banner, author info, and publication date.
2) **Similar Blogs**
   - Displays a list of similar blogs based on tags.

MinimalBlogPost

The MinimalBlogPost component is designed to display a concise summary of blog posts on the homepage, emphasizing minimalistic design. It serves as a quick preview for users to navigate to full blog content.

Features

1) **Compact Design**:
   - Displays the blog title, index number, and publication date in a simple layout.
2) **Interactive Navigation**:
   - Clicking on the component navigates to the corresponding blog page using react-router-dom.
3) **Responsive Elements**:
   - Handles data dynamically with placeholders for profile details.
   - Displays only essential blog information for quick scanning.

Connections to Other Modules

- Date Utility: Uses getDay from a shared common/date utility module to format publication dates.

Loader

The Loader component provides a visual indicator for loading states, enhancing user experience by notifying them that content is being processed or fetched.

Features

1) **Animated Spinner:**

- Displays a spinning SVG animation to indicate loading progress.

Connections to Other Modules

- Homepage and Blog Pages: Used while fetching data such as blog posts or user details.

AnimationWrapper

The AnimationWrapper component enhances the visual appeal of the application by wrapping elements with smooth animations, creating a dynamic and engaging user interface.

Features

1) **Dynamic Animations:**

- Utilizes framer-motion to provide entry and exit animations for child components.

Connections to Other Modules

- **Homepage and Blog Pages**: Adds smooth animations to elements like blog cards or headers.

InPageNavigation

The InPageNavigation component provides dynamic navigation system within a page, improving user experience by allowing content switching without page reloads.

Features

1) **Dynamic Tab Navigation:**

- Displays multiple navigation tabs based on provided routes.

Connections to Other Modules

- **Homepage and Blog Pages**: Enables in-page navigation for switching between sections like "Latest Blogs" or "Top Rated."

NoDataMessage

The NoDataMessage component provides a user-friendly message when no data is available, with a informative placeholder.

1) **Customizable Message:**

- Accepts a message prop, allowing dynamic text based on the context (e.g., "No blogs available" or "No activity to display").

Connections to Other Modules

- **Blog Pages**: Displays when no blog posts are available or loaded.

LoadMoreDataBtn

The LoadMoreDataBtn component allows users to load additional data (e.g., blog posts, activities) when they scroll or request more. It provides a button that triggers data fetching for the next set of results.

Features

1) **Conditional Rendering:**

- The button only appears if more data is available (i.e., the total number of documents exceeds the number of currently loaded results).

2) **Data Fetching:**

- On click, it triggers a function (fetchDataFun) to fetch more data**.**

Connections to Other Modules

- Home Page: Appears on the home page after login to load additional content such as blog posts or user activity.

getDay

The getDay function formats a given timestamp into a user-friendly date format (e.g., "1 Jan") by extracting the day of the month and the month name.

Features

1. **Date Formatting**:
    - Converts a timestamp into a readable date format using arrays for months and days.

Connections to Other Modules

- **Blog and Profile Pages**: Used for displaying the publication or activity date in blog posts.lt

filterPaginationData

The filterPaginationData function handles data fetching and pagination logic. It either appends new data to an existing array or fetches the total number of documents from the server for a fresh set of data.

Features

1. **Pagination Management**:
    - Supports both appending new data and retrieving a new data set from the server.

Connections to Other Modules

- **Homepage**: Integrates with the homepage to fetch and display paginated blog data.

Editor

The Editor component manages the blog creation process, allowing users to either edit or publish a blog post. It handles the state of the blog content and

controls the editor's user interface flow (either in the editing mode or the publishing form).

Features

1) **State Management:**
   - Manages the blog structure and editor state (editor or publish form) using React hooks.
2) **Conditional Rendering:**
   - Conditionally renders the **BlogEditor** for editing or **PublishForm** for publishing based on the editorState.

BlogEditor

The BlogEditor module enables users to create and edit blog posts. The module integrates with the AWS S3 service for image uploads and handles data interactions with a backend API to save the blog post.

Features

1) **Image Upload**:
   - Allows users to upload a banner image for the blog, with integration for image hosting via AWS.
2) **Dynamic Title Input**:
   - Supports a dynamic text input for the blog title, which auto-resizes based on the input length.
3) **Text Editor**:
   - Implements EditorJS for rich-text content creation, enabling users to write and format the blog post content.
4) **Save Draft and Publish Options**:
   - Users can save their blog as a draft or publish it. The draft is saved locally and can be retrieved later.
5) **Backend Interaction**:
   - Sends the blog data to a backend API for saving drafts and publishing posts.
6) **Toasts for Notifications**:
   - Displays notifications for successful or failed actions, such as saving drafts or uploading images.

BlogContent

The BlogContent module is responsible for rendering various types of content blocks within a blog, such as paragraphs, headers, images, quotes, and lists.

Features

1) **Dynamic Content Rendering:**
   - Supports multiple content types (paragraph, header, image, quote, and list) based on the data provided.
2) **Image and Caption Handling**
   - Displays images with optional captions.
3) **List Styling**
   - Supports both ordered and unordered lists.

Tools

The tools module configures various editor tools for the blog editor, enabling content creation with embedded media, lists, images, headers, quotes, and inline formatting.

Features:

1) **Embed Tool**
   - Allows embedding external content (e.g., videos, tweets).
2) **List Tool**
   - Supports creating ordered and unordered lists with inline toolbar functionality.
3) **Image Tool**
   - Provides image uploading through both file and URL methods.
4) **Header Tool**
   - Lets users add headings with different levels, supporting a placeholder for easy customization.
5) **Quote Tool**
   - Enables inserting and styling quotes with inline editing.

6) **Marker Tool**
   - Adds highlighted text (e.g., for emphasis).
7) **InlineCode Tool**
   - Allows inline code formatting for technical content.

PublishForm

The PublishForm module allows users to complete and publish a blog after creating it in the BlogEditor. This form serves as the final step where users can preview their blog, add a title, description, and tags, then publish the blog to the platform.

Features

1) **Preview Section**
   - Displays a preview of the blog with its **title, description, and banner image**.
2) **Blog Title and Description**
   - Users can modify the **blog title and description** before publishing.
3) **Tags Management**
   - Users can add **tags** to categorize the blog
4) **Publish Button**
   - Users can publish their blog once all required fields are filled, which triggers the blog data to be saved to the backend.

BlogInteraction

The BlogInteraction module enables users to interact with a blog post by liking the blog and sharing it on Twitter. It displays the like count and provides a way for users to share the blog on social media.

Features:

1) **Like Button**
   - Allows users to **like or unlike** the blog. The like count is updated dynamically based on the user's action.
2) **Twitter Sharing**
   - Provides a button for sharing the blog on Twitter with a preformatted message and link to the blog post.
3) **Dynamic Like Count**
   - Updates the total number of likes when a user likes or unlikes the blog.

Tag

The Tag module allows users to edit and delete tags associated with a blog post. It provides a way to interactively manage tags in the blog editor.

Features:

1) **Editable Tags**
   - Tags are initially displayed as text but can be made editable by clicking on them, allowing users to modify the tag name.
2) **Tag Deletion**
   - Users can **delete tags** by clicking the delete icon next to each tag.

ManageBlogs(not integrated)

The ManageBlogs module provides an interface for logged-in users to manage their blog posts. It allows users to view and search through both published and draft blogs, and perform actions such as loading more data and navigating between tabs for each category.

Features:

1) **Search Functionality**
   - Allows users to search for blogs by **title** or **content.**
2) **Tab Navigation**
   - Displays two tabs — **"Published Blogs"** and **"Drafts"** — allowing users to switch between them.
3) **Pagination**
   - Supports pagination for viewing large sets of blogs with the option to load more content.

Connections with Other Modules:

- Uses components like **ManageDraftBlogPost** and **ManagePublishedBlogCard** to display each blog and handle blog-specific actions.

ManagePublishedBlogCard and ManageDraftBlogPost

(not integrated)

The ManagePublishedBlogCard and ManageDraftBlogPost modules provide functionality for users to manage their published and draft blogs.

Features:

1) **Blog List**
   - Displays a list of published and draft blogs with titles, descriptions, and metadata.
2) **Stats Toggle**
   - Allows users to toggle visibility of blog statistics (views, comments, etc.).
3) **Edit & Delete**
   - Users can edit or delete blogs directly from the list.

PageNotFound

The PageNotFound module is designed to display a user-friendly 404 error page when a user attempts to navigate to a non-existent or unavailable page.

Features:

1) **404 Error Message**
   - Displays a clear message indicating that the requested page was not found.
2) **Navigation**
   - Provides a link back to the home page for easy redirection.