ChicAI : Smart Wardrobe

GitHub Repository Link : ChicAI

ChicAI is a smart wardrobe app designed to simplify outfit selection and elevate personal style through features like digital wardrobe management, weather-based and event-specific outfit suggestions, laundry tracking, trend-based shopping recommendations, and customizable dashboards. It allows users to catalog their wardrobe, mix and match outfits, and share them with others. With its seamless integration of user preferences, calendar events, and weather data, ChicAI stands out as a comprehensive and free personal stylist app for all user types.

Features:
Photograph and catalog your wardrobe.
Receive three personalized daily outfit suggestions, including weather and event-based recommendations.
Track clothes marked as "in laundry" and save favorite outfits.
Integrates with event calendars for tailored planning.
Completely free with unique features like laundry tracking and personalized shopping links.

Tech Stack:
Frontend: ReactJS
Backend: Node.js with Express.
Database: MongoDB.
Cloud Services: AWS S3
APIs: OpenWeatherMap API for weather integration, Google Calendar API for event synchronization.
AI/ML: OpenAI API.
Authentication: Firebase Authentication for secure user login.
Notifications: Push notifications through Firebase Cloud Messaging.

List of APIs

Request Type	Endpoint	Query Parameter	Description	Example input	Example Output
POST	/register	None	Allows user registration with validation for username and password criteria.	{ "name": "John Doe", "email": "user@example.com", "password": "Password1" }	{ "message": "Registration successful" }
POST	/login	None	Allows users to log in using their email and password.	{ "email": "user@example.com", "password": "Password1" }	{ "token": "auth_token", "message": "Login successful" }

GET	/profile	userId	Fetches profile details of the logged-in user.	userId=123	{ "name": "John Doe", "dob": "1998-01-01", "gender": "Male", "occupation": "Software Developer" }
PUT	/profile	None	Updates user profile details.	{ "name": "Jane Doe", "gender": "Female", "occupation": "Designer" }	{ updated profile payload }
POST	/items	None	Adds a clothing item to the wardrobe with optional tags and weather categorization.	{ "title": "Blue Jeans", "tags": ["Casual"], "weatherTag": ["Winter"], "image": "<S3 url>" }	{ added item payload }
GET	/items	filter, search	Retrieves all items or filtered items in the wardrobe based on search criteria or filters.	filter=Winter,Casual&search=Jeans	[{ "title": "Blue Jeans", "tags": ["Casual"], "weatherTag": ["Winter"] }]
GET	/item/:id	id	Retrieves details of a specific clothing item.	/item/1	{ "title": "Blue Jeans", "tags": ["Casual"], "weatherTag": ["Winter"], "description": "Comfortable denim jeans" }
PUT/Patch	/item/:id	id	Updates details of a specific clothing item.	{ "tags": ["Formal"], "description": "Updated description" }	{ updated item payload }
DELETE	/item/:id	id	Deletes a clothing item from the wardrobe.	/api/wardrobe/item/1	{ "message": "Item deleted successfully" }
POST	/laundry	None	Tags items as “in-laundry.”	{ "items": [1, 2, 3] }	{ "message": "Items tagged as in-laundry" }
GET	/laundry	None	Retrieves a list of items marked as “in-laundry.”	None	[{ "title": "Blue Jeans", "status": "in-laundry" }]
PUT/Post	/laundry	None	Unmarks items as “in-laundry.”	{ "items": [1, 2, 3] }	{ "message": "Items unmarked as in-laundry" }
POST	/events	None	Adds or links events for outfit suggestions.	{ "eventName": "Party", "date": "2024-12-25" }	{ "message": "Event added successfully" }
GET	/suggestions/outfits	event, weather	Fetches outfit suggestions based on events	weather=Rainy	[{ "title": "Raincoat", "tags": ["Rainy"], "weatherTag": "Rainy" }]

			or weather.		
POST	/feedback	None	Submits user feedback.	{ "feedback": "Great app!" }	{ "message": "Feedback submitted successfully" }
POST	/suggestions/:id/save	None	Save/Unsave an Outfit suggestion to view later	item/1/save=True	[{"id":1, "title": "Blue Jeans", "tags": ["Casual"], "weatherTag": ["Winter"], "description": "Comfortable denim jeans" }, {"id":3, "title": "Blue Jeans", "tags": ["Casual"], "weatherTag": ["Winter"], "description": "Comfortable denim jeans" }]
GET	/save	None	Fetches a list of saved outfit suggestions		

Product Requirements

User Stories with Acceptance Criteria:

User Roles:

- End User
- DB Admin
- DB Super Admin

1. As an end-user, I should be able to register, so that I can use the application.
   Acceptance Criteria:
   - The username should have at least 6 characters
   - Sign-up should be authenticated using email and password
   - Password must contain letters and numbers

2. As an end-user, I should be able to log in, so that I can use my dashboard
   Acceptance Criteria:
   - The username should have at least 6 characters
   - Login should be authenticated using a password
   - Password must contain letters and numbers

3. As an end-user, I should be able to edit my profile details, for customized suggestions.
   Acceptance Criteria:
   - The dashboard should have a profile tab
   - The user should be able to edit and save personal details (Name, DOB, Gender, Occupation)
   - Upon saving, changes should be reflected on the Profile Page

4. As an end-user, I want to log my entire wardrobe with photographs or manually so that I can easily view all my clothing options without rummaging through my wardrobe.
   Acceptance Criteria:
   - Users should be able to add items by uploading photographs and manually inputting details
   - Each clothing item should have the option to add tags to the item to better categorize them
   - If a photograph is uploaded, the system should prompt the user to add or edit a description before saving the item to the database.
   - Each clothing item should have the option to be tagged as either summer or winter to assist with weather-based categorization.

5. As an end-user, I want to view all items currently in my wardrobe, so that I know which clothes I own.
   Acceptance Criteria:
   - The user should be able to view a list of all clothing items stored in the digital wardrobe
   - Each item should be displayed with its image, title, description, and category tags

6. As an end-user, I want to read/update details about a specific item, so that I can add/change the description or category tags for the item.
   Acceptance Criteria:
   - User should be able to read/update a list of all clothing items currently in the digital wardrobe
   - Each item should be displayed with its image, title, and description
   - The user should be able to add or delete category tags from the item

7. As an end-user, I want to be able to delete details about a specific item or the item itself, so that I can keep my digital wardrobe updated with items in my physical wardrobe.
   Acceptance Criteria:
   - The user should be able to delete items
   - Deleted items should not be displayed

8. As an end-user, I want to filter and search through the list of clothes based on their category, so I can easily find what I am looking for.
   Acceptance Criteria:
   - The list view should have a search option
   - The list view should have multiple filters based on item category, color, weather tag, etc
   - Upon applying filters, the updated filtered list should be displayed

9. As an end-user, I would like to categorize outfits with the use of tags.
   Acceptance Criteria:
   - Users should be able to add or edit tags
   - Tag title length will be of length 24 characters maximum
   - Users should be able to filter based on tags
   - Tags should be displayed in list views

10. As an end-user, I want to add/view/update/delete items, so that I can keep my digital wardrobe updated when I throw items.
    Acceptance Criteria:
    - Upon selecting a certain Item, allow users to edit the item
    - The user should be able to perform read/update/delete operations on the selected item

11. As an end-user, I want to mark items I add to laundry so that I don't get suggestions for in-laundry items.
    Acceptance Criteria:
    - On selecting an outfit, manually tag “in-laundry”
    - Do not suggest items marked In-laundry tagged
    - Should display “in-laundry” and “in-wardrobe” items separately

12. As an end-user, I would like to keep track of what's in what's in the laundry so that I can plan my laundry better.

Acceptance Criteria:

- The dashboard should have a Laundry navigation option
- Upon navigation, a list of items tagged as "in-laundry" should display
- Each item should be displayed with its image, title, and description

13. As an end-user, I would like to unmark what's in the laundry after I do my laundry, so the clothes can be back in rotation.

    Acceptance Criteria:

    - User should be able to select individual or multiple items and change the "in-laundry" tag to unmark items
    - Upon un-marking items as "in-laundry", the items should be available for suggestion again
    - Items unmarked, should not show up in the Laundry tab

14. As an end-user, I want to know what to wear based on my upcoming events, so that I don't have to plan outfits for events ahead of time.

    Acceptance Criteria:

    - User should be able to link their event calendar or manually add future events
    - On the day of the event, the user should get outfit suggestions based on the type of event added

15. As an end-user, I would like to mix and match different wardrobe items to create outfits that can be viewed and selected for another day.

    Acceptance Criteria:

    - Ability to manually combine different wardrobe items to create an outfit (group them)
    - Ability to view all created outfits

16. As an end-user, I would like to save the suggestions provided, so that I can go through them when needed.

    Acceptance Criteria:

    - Outfit suggestions should have a toggle button to save the given suggestion
    - The dashboard should display a Saved Outfits navigation
    - Upon navigation, a list of outfits saved by the user should display
    - Each saved item should be displayed with its image, title, and description

17. As an end-user, I would like the app to suggest outfits based on the weather forecast so that I can dress appropriately.

    Acceptance Criteria:

    - Fetch weather data from the weather API
    - Suggest Outfits that match the appropriate weather tag

18. As an end-user, I would like to receive suggestions based on current fashion trends, so that it makes shopping easy.

    Acceptance Criteria:

- Fetch trendy clothing products from shopping apps using API
- Recommend products to buy when desired

19. As an end-user, I would like to share my outfit suggestions with friends and family so that I can get their opinions.
    Acceptance Criteria:
    - Each Outfit suggestion should have a share button
    - Share button should generate a link to the outfit that can be publicly accessed
    - The link should display an image of the shared outfit
    - The generated link should expire within 1 hour of being generated

20. As an end-user, I would like to add notes to each clothing item so that I can keep track of special care instructions.
    Acceptance Criteria:
    - Each item in the digital wardrobe has a "Notes" section
    - Users can add, edit, and delete notes for any wardrobe item
    - The notes field should support text input of at least 512 characters
    - Notes are displayed when viewing item details

21. As an end-user, I want to be able to submit feedback about the app so that I can share my experience, report issues, or suggest improvements
    Acceptance Criteria:
    - The feedback form should include fields for entering feedback text
    - Feedback text should be limited to 512 characters
    - Upon submission, the user should receive a confirmation message acknowledging receipt of the feedback

22. As an admin user, I would like to view and manage customer feedback so that I can monitor user satisfaction
    Acceptance Criteria:
    - The admin should be able to view all customer feedback in list/grid format
    - The feedback should include the user's name, feedback details, and the date submitted
    - The admin should be able to sort and filter feedback by date, or user.
    - The admin should be able to mark feedback as "addressed" or "pending"
    - The admin should be able to delete inappropriate feedback

23. As an admin user, I want to add/update/delete tags, so that they are synchronous with new app updates and features
    Acceptance Criteria:
    - The admin should be able to perform CRUD operations on tags
    - The changes should be reflected to end-users upon implementation

24. As an admin user, I want to extract data about users and user-type of users so that I can analyze the app's usage

Acceptance Criteria:

- The admin should be able to extract all users in the form of a CSV
- Data export should have a date range filter for both viewing and exporting

25. As an admin user, I want to manage all users and their profile information
    Acceptance Criteria:
    - The admin should be able to view all users, including their profile information
    - The admin should be able to delete any users
    - The admin should be able to bulk-delete users

26. As a super admin user, I want to create, update, view, and delete admin accounts so that I can manage the platform's administrative team.
    Acceptance Criteria:
    - The super admin can create new admin accounts
    - The super admin can update admin details
    - The ability to create, update, view, and delete admins
    - The super admin can delete admin accounts

27. As a super admin user, I want to have the same functionalities as a regular admin, in addition to my super admin capabilities, so that I can perform regular admin tasks when needed.
    Acceptance Criteria:
    - The super admin can access all admin functionalities, such as managing customer feedback, viewing users, and extracting data.
    - The super admin retains additional super admin capabilities

ChicAI is a smart wardrobe app designed to simplify outfit selection and elevate personal style through features like digital wardrobe management, weather-based and event-specific outfit suggestions, laundry tracking, trend-based shopping recommendations, and customizable dashboards. It allows users to catalog their wardrobe, mix and match outfits, and share them with others. With its seamless integration of user preferences, calendar events, and weather data, ChicAI stands out as a comprehensive and free personal stylist app for all user types.

Features:

- Photograph and catalog your wardrobe.
- Receive three personalized daily outfit suggestions, including weather and event-based recommendations.
- Track clothes marked as “in laundry” and save favorite outfits.
- Integrates with event calendars for tailored planning.
- Completely free with unique features like laundry tracking and personalized shopping links.

Tech Stack:

- Frontend: ReactJS
- Backend: Node.js with Express.
- Database: MongoDB.
- Cloud Services: AWS S3
- APIs: OpenWeatherMap API for weather integration, Google Calendar API for event synchronization.
- AI/ML: OpenAI API.
- Authentication: Firebase Authentication for secure user login.
- Notifications: Push notifications through Firebase Cloud Messaging.