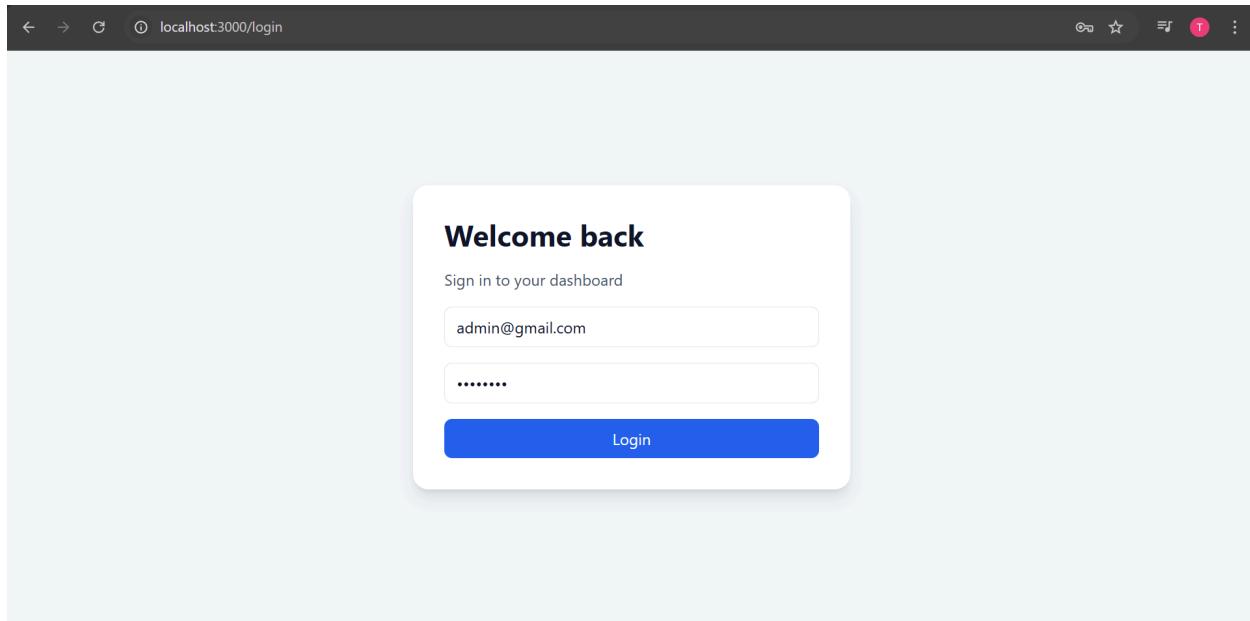


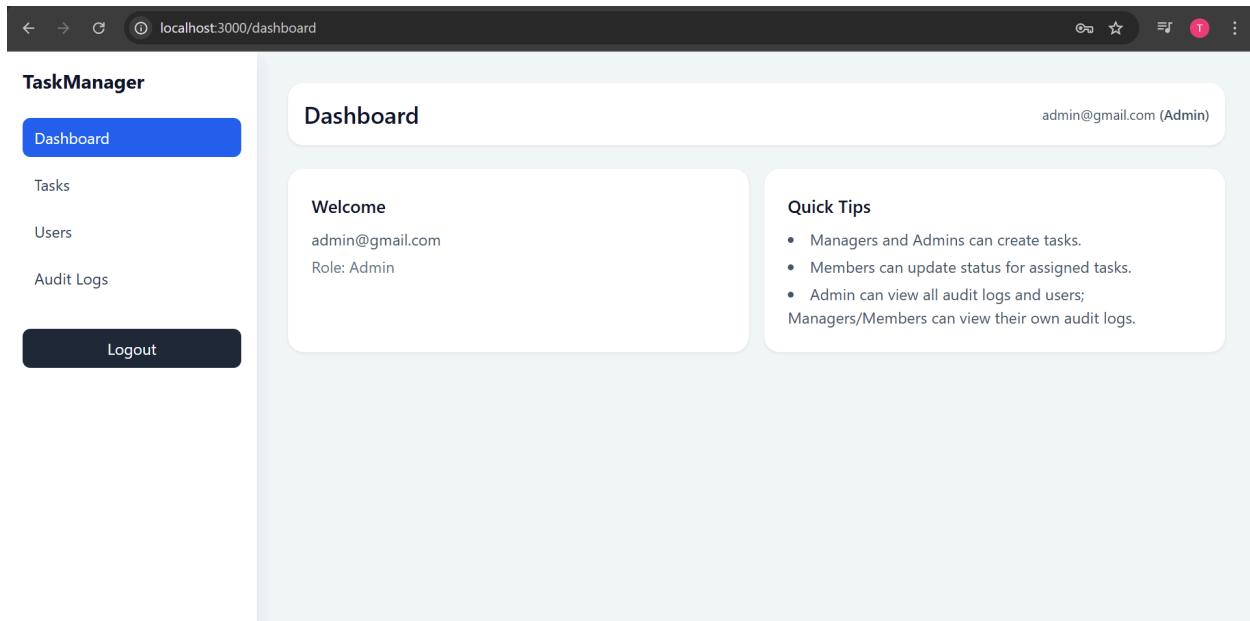
Admin

Login Page



A screenshot of a web browser showing the login page for an admin application. The URL in the address bar is `localhost:3000/login`. The page features a central modal with a white background and rounded corners. The title "Welcome back" is at the top. Below it is the instruction "Sign in to your dashboard". There are two input fields: the first contains "admin@gmail.com" and the second contains "*****". At the bottom is a large blue "Login" button.

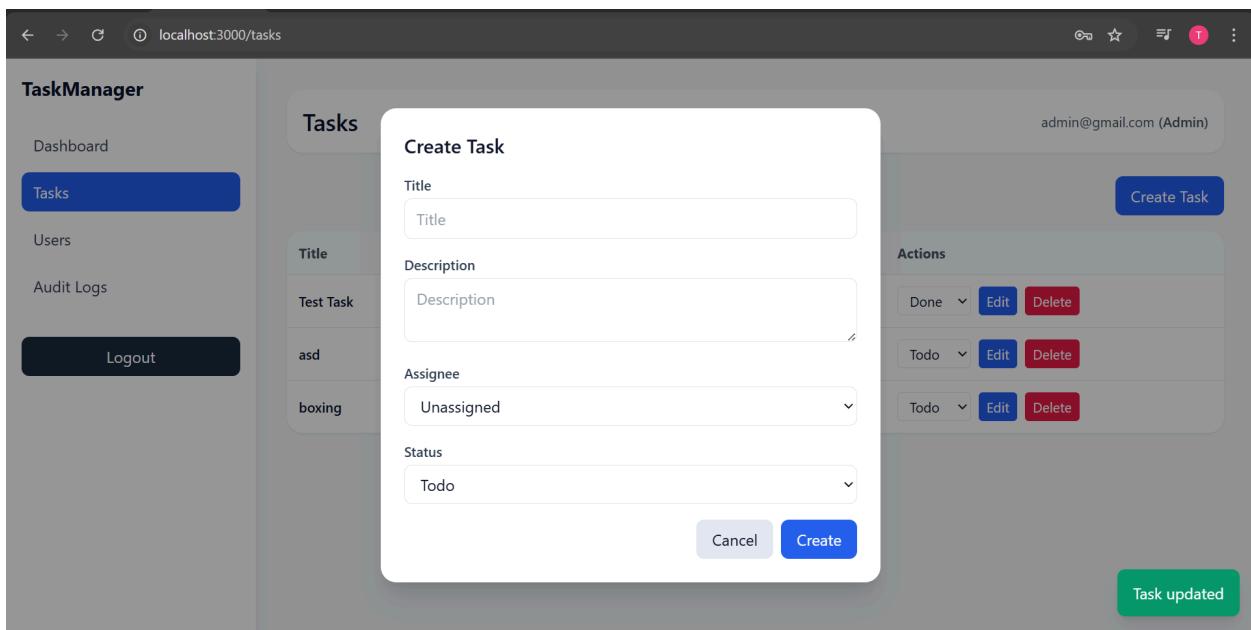
Dashboard



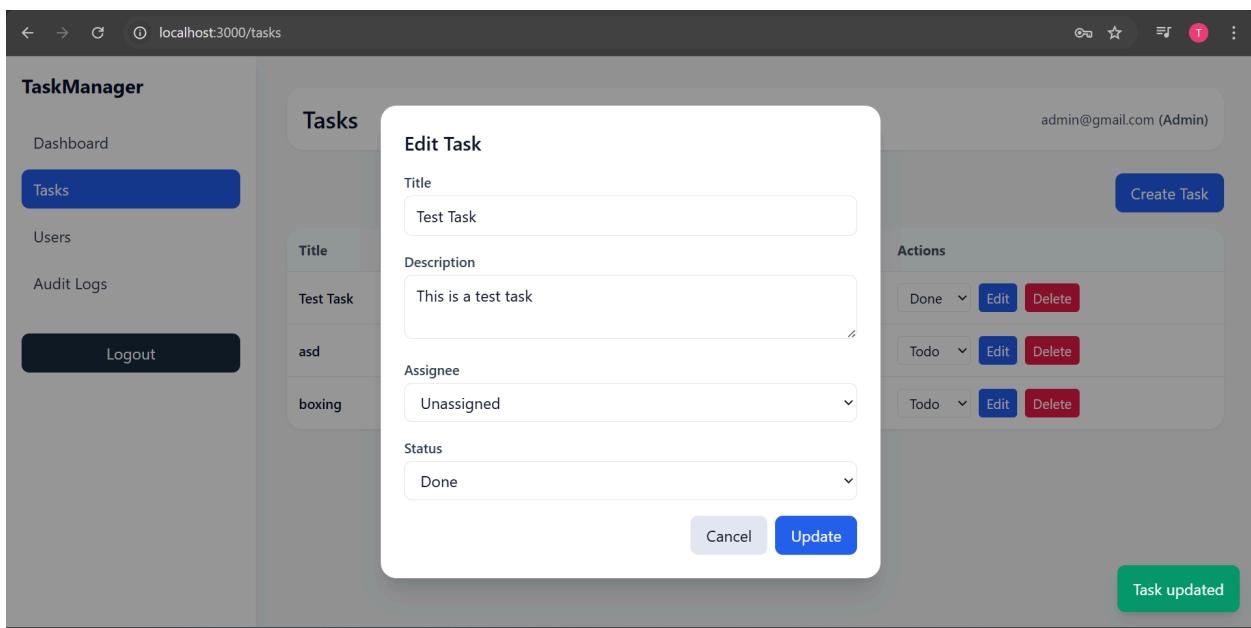
A screenshot of a web browser showing the dashboard for the admin application. The URL in the address bar is `localhost:3000/dashboard`. On the left, a sidebar titled "TaskManager" has a "Dashboard" button highlighted in blue. Other menu items include "Tasks", "Users", and "Audit Logs". At the bottom of the sidebar is a "Logout" button. The main content area has a header "Dashboard" and a user profile "admin@gmail.com (Admin)". Below this is a "Welcome" section showing "admin@gmail.com" and "Role: Admin". To the right is a "Quick Tips" section with the following bullet points:

- Managers and Admins can create tasks.
- Members can update status for assigned tasks.
- Admin can view all audit logs and users; Managers/Members can view their own audit logs.

Create task Model



Edit task Model



Manage user

The screenshot shows a web browser window for the TaskManager application at localhost:3000/users. The left sidebar has a 'TaskManager' header and links for Dashboard, Tasks, Users (which is highlighted in blue), Audit Logs, and Logout. The main content area has a 'Users' header and a table with columns ID, Email, and Role. The table contains six rows of user data:

ID	Email	Role
1	admin@example.com	Admin
2	tenzin@example.com	Member
3	dorji@gmail.com	Admin
4	tenzin@gmail.com	Member
5	tengyel@gmail.com	Manager
6	admin@gmail.com	Admin (you)

Audit logs

The screenshot shows a web browser window for the TaskManager application at localhost:3000/audit. The left sidebar has a 'TaskManager' header and links for Dashboard, Tasks, Users, Audit Logs (which is highlighted in blue), and Logout. The main content area has an 'Audit Logs' header and a table with columns ID, User, Action, Entity, and Timestamp. The table contains ten rows of audit log data:

ID	User	Action	Entity	Timestamp
11	6	UPDATE	Task	2/16/2026, 5:51:58 PM
10	5	UPDATE	Task	2/12/2026, 8:44:21 AM
9	5	UPDATE	Task	2/12/2026, 8:44:17 AM
8	5	UPDATE	Task	2/12/2026, 8:44:14 AM
7	3	CREATE	Task	2/12/2026, 8:27:42 AM
6	3	UPDATE	Task	2/12/2026, 8:27:18 AM
5	3	UPDATE	Task	2/12/2026, 8:27:14 AM
4	3	UPDATE	Task	2/12/2026, 8:27:11 AM
3	3	CREATE	Task	2/12/2026, 8:20:07 AM

Manager

The screenshot shows a web browser window for the URL `localhost:3000/tasks`. The title bar includes standard navigation icons. The main content area has a header "TaskManager" and a sub-header "Tasks". On the right, the email "manager@gmail.com (Manager)" is displayed. A "Create Task" button is located in the top right corner of the main content area. The central part of the screen is a table listing tasks:

Title	Description	Status	Assignee	Actions
Test Task	This is a test task	Done	Unknown	Done ▾ Edit Delete
asd	asd	Todo	Unassigned	Todo ▾ Edit Delete
boxing	oka	Todo	Unassigned	Todo ▾ Edit Delete

Member

The screenshot shows a web browser window for the URL `localhost:3000/tasks`. The title bar includes standard navigation icons. The main content area has a header "TaskManager" and a sub-header "Tasks". On the right, the email "member@gmail.com (Member)" is displayed. A "Create Task" button is located in the top right corner of the main content area. The central part of the screen is a table listing tasks:

Title	Description	Status	Assignee	Actions
Test Task	This is a test task	Done	Unknown	Done ▾

Viewer

The screenshot shows a web application interface. At the top, there's a header bar with a back arrow, forward arrow, refresh button, and a URL field containing "localhost:3000/tasks". To the right of the URL are several small icons. Below the header is a sidebar on the left labeled "TaskManager" with three buttons: "Dashboard", "Tasks" (which is highlighted in blue), and "Logout". The main content area is titled "Tasks" and contains a table with four columns: "Title", "Description", "Status", and "Assignee". There are three rows of data:

Title	Description	Status	Assignee
Test Task	This is a test task	Done	Unknown
asd	asd	Todo	Unassigned
boxing	oka	Todo	Unassigned

Postman

Register

The screenshot shows the Postman application interface. At the top, there's a header bar with tabs for different API endpoints. The active tab is "POST http://127.0.0.1:8000/api/auth/register". The main workspace shows a "POST" request to "http://127.0.0.1:8000/api/auth/register". The "Body" tab is selected, showing the following JSON payload:

```
1 { "email": "viewer@gmail.com", "password": "password", "role": "Viewer" }
```

Below the request, the "Test Results" section displays the response details. It shows a status of "200 OK", a time of "205 ms", and a size of "218 B". The response body is shown in "Pretty" format:

```
1 {  
2   "email": "viewer@gmail.com",  
3   "id": 9,  
4   "role": "Viewer",  
5   "created_at": "2026-02-16T18:04:07.552451"  
6 }
```

Login

The screenshot shows the Postman application interface. At the top, there are several tabs: POST http://127.0.0.1:8000/api/, POST http://127.0.0.1:8000/api/, GET http://127.0.0.1:8000/api/, and POST http://127.0.0.1:8000/api/. The active tab is 'POST http://127.0.0.1:8000/api/auth/login'. The request method is set to 'POST' and the URL is 'http://127.0.0.1:8000/api/auth/login'. The 'Body' tab is selected, showing a 'form-data' payload with two fields: 'username' (value: admin@gmail.com) and 'password' (value: password). Below the body, the 'Test Results' tab is visible, showing a successful response with status 200 OK, time 217 ms, and size 304 B. The response body is a JSON object containing an access token and a token type.

```
1 "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWJlIjoiJhZG1pbkBnbWFpbC5jb20iLCJleHAiOjE3NzEyNjg4MzB9.Xj0pvMZBsImr7NBC_7uZ007kclhBvXGK9wV7hJc0BQQ",
2 "token_type": "bearer"
3
4 }
```

Swagger UI

The screenshot shows the Task Management API documentation generated by Swagger UI. The title is 'Task Management API' with version 0.1.0 and a green 'GAS3.1' badge. The interface includes a search bar, a download button, and an 'Authorize' button. The API is organized into several sections: 'auth' (with POST /api/auth/register and POST /api/auth/login), 'users' (with GET /api/users/me, GET /api/users, and PUT /api/users/{user_id}/role), 'tasks' (with GET /api/tasks, POST /api/tasks, PUT /api/tasks/{task_id}, and DELETE /api/tasks/{task_id}), 'audit' (with GET /api/audit), and 'default' (with GET /). Each endpoint is shown with its method, URL, and a brief description.

The screenshot displays a Swagger UI interface for a RESTful API. At the top, there is a navigation bar with links for "Home", "Documentation", "Schema", "Logs", and "Help". Below the navigation, the main content area is organized into sections:

- API Endpoints:** A list of endpoints grouped by their corresponding API groups:
 - audit**:
 - DELETE /api/tasks/{task_id}** Remove Task
 - default**:
 - GET /api/audit** List Audit Logs
 - GET /** Health
- Schemas:** A list of schema definitions:
 - AuditLogOut** > Expand all object
 - Body_login_api_auth_login_post** > Expand all object
 - HTTPValidationError** > Expand all object
 - TaskCreate** > Expand all object
 - TaskOut** > Expand all object
 - TaskStatus** > Expand all string
 - TaskUpdate** > Expand all object
 - Token** > Expand all object
 - UserCreate** > Expand all object
 - UserOut** > Expand all object
 - UserRoleUpdate** > Expand all object
 - ValidationError** > Expand all object

Dockerization

The screenshot shows a browser window with two tabs and a Docker Desktop interface below it.

- Task Management Dashboard:** The left tab shows a "Welcome back" message and a sign-in form with the email "admin@gmail.com".
- Docker Desktop:** The right tab displays the Docker desktop interface. The sidebar includes "Ask Gordon", "Containers", "Images", "Volumes", "Kubernetes", "Builds", "Models", and "MCP Toolkit (BETA)". The main area shows container statistics: "Container CPU usage" at 0.10% / 2000% (20 CPUs available) and "Container memory usage" at 116.05MB / 7.39GB. A "Show charts" link is also present. Below this, a table lists running containers with columns for Name, Container ID, Image, Port(s), CPU (%), Last std, and Actions. One container named "tcell-assignmen" is listed.