**T-ming**

**Slide 1: Intro**

**Divya:**  
Hello everyone! I’m Divya

**Ghanshyam:**  
And I’m Ghanshyam.  
Today, we’re here to introduce something that we’ve built—**T-ming**, a team-based project management system designed to make teamwork and task management seamless.

**Slide 2: T-ming, in a nutshell**

**Divya:**  
T-ming is **powerful, scalable, team-based project management system and built using Node.js, Express, MongoDB, and React**. It’s designed for **real-world B2B needs**, offering **Google Sign-In, workspace management, project tracking, task collaboration, role-based permissions**, and much more.   
Whether you're a **freelancer, a startup, or a large organization**, managing projects efficiently is always a challenge. T-ming helps structure your workflow and **keeps everything organized**.

**Slide 3: The key features**

**Ghanshyam:**

Let me tell you about some features our software provides:

Let’s start with **authentication**—because security matters. Users can log in using **Google Sign-In** or their **email and password**. This ensures that accounts stay protected while making access seamless.

**Divya:**  
Now, once logged in, users can create and manage multiple **workspaces**. Each workspace acts as a separate environment where projects, tasks, and members are neatly organized.

**Ghanshyam:**  
Within a workspace, users can manage **projects and tasks**. Projects help in **structuring work**, and tasks are the core of getting things done. Tasks can be **created, read, updated, or deleted**—the standard CRUD operations.   
To keep things even more structured, tasks come with additional attributes—**status, priority level, and assignee**—so that teams can track progress efficiently.

**Divya:**  
Security and data integrity are a big part of T-ming. We use **cookie-based session management**, keeping users securely logged in, and allowing them to **log out anytime** while ensuring proper session termination.

**Ghanshyam:**  
As for the **tech stack**, T-ming is built using **MERN**—MongoDB, Express, React, and Node.js—ensuring a smooth and scalable development experience. We also use **TypeScript** for strong typing, making our codebase more reliable. And for easier testing, we can even **seed test data** into the system.

**Divya:**  
Moving on to **user roles and access**—every workspace has three types of users: **Owners, Admins, and Members**. Owners and Admins can **invite new members**, keeping the team structured.

**Ghanshyam:**To make navigation easy, we also have **smart search and filtering**. Users can search by **status, priority, and assignee**, helping them find tasks instantly.

**Divya:**  
And for **advanced features**, we have an **analytics dashboard** to track project progress and **pagination & load more** to handle large data sets smoothly.

**Slide 4: The tech stack**

**Ghanshyam:**  
Let’s talk about what powers T-ming under the hood. It’s built using the **MERN stack**—MongoDB, Express, React, and Node.js—providing a full-stack, seamless experience.   
For **backend development**, we use **Node.js** for its **scalability and efficiency**, and **MongoDB with Mongoose** to make data handling flexible and reliable.

**Divya:**On the **frontend**, we leverage **React.js** to make the UI **dynamic and component-driven**, and **Vite.js** to ensure fast development with optimized performance.

**Ghanshyam:**  
Security is handled through **Google OAuth**, making the authentication process safe and hassle-free. We also use **TypeScript**, which keeps the code more reliable and maintainable.

**Divya:**And to make everything look modern and user-friendly, we use **Tailwind CSS** for styling and **Shadcn UI components** to enhance the UI experience.

**Slide 5: The DFDs**

**Ghanshyam:**  
Now, let’s break down how T-ming works at a **system level** with **Data Flow Diagrams**.

**Divya:**  
At the highest level, the system has six key processes:

1. **User Authentication** – Ensuring secure login and registration.
2. **Workspace Management** – Users create workspaces, and the data is stored in the database.
3. **Project Management** – Adding, updating, and deleting projects in real-time.
4. **Task Management** – Creating and modifying tasks, with all updates saved in the database.
5. **Role Assignment** – Owners assign roles and invite members, defining access levels.
6. **Database Interaction** – Every action interacts with the database for data retrieval and storage.

**Slide 6: The Auth level 1**

**Ghanshyam:**  
Let’s take a **closer look at authentication**, where users can **register or log in** using their **credentials or Google Authentication**.

When a new user **registers**, their credentials are **stored securely** in the database.

And when logging in, the system **checks these credentials** against the database to **validate the user**. If everything is correct, the system **verifies the user’s existence** and **loads their data**.

Behind the scenes, the **database manages all interactions** related to **user registration, authentication, and data retrieval**.

**Slide 7: The Workspace management**

**Divya:**  
Now, let’s talk about **workspace management**. Users can **create a workspace** with a title and description.   
Once created, the workspace **gets stored in the database**. Users can later **edit or delete it**, and the database updates accordingly.

**Slide 8: The Project management**

**Ghanshyam:**  
Similarly, we have **project management**. Users create projects, store them in the database, and can later **modify or delete them**.

**Slide 9: Task management**

**Divya:**  
Tasks follow the same flow but with added details like **title, description, assignee, due date, status, and priority**.

And permissions play a role here—**users with the right access can edit or delete tasks and projects.**

**Slide 10: The User management**

**Ghanshyam:**Talking about the **roles,** we have **User management**, another key part of the system. The **workspace owner** controls who get what role, and based on that the access level is also specified.  
Owners can **invite users**, and the system updates the database to reflect new members.   
They can also **assign roles**—Admin or Member—granting different permissions.

**Slide 11: The ERD**

**Divya:**  
To better understand how everything connects, we have the **Entity-Relationship Diagram (ERD)**.   
Our system consists of multiple entities, each with its own attributes:

* **User** – Stores essential details like **UserID, Name, Email, Profile Picture, and more**.
* **Account** – Manages authentication through different providers like **Google or Email**.
* **Workspace** – Each workspace has an ID, name, and invite code.
* **Project** – Belongs to a workspace and includes details like **ProjectID, Name, and CreatedBy**.
* **Task** – Each task has attributes like **TaskID, Title, Status, Priority, and Due Date**.

**Ghanshyam:**  
And how do they connect? So, the relationship is like

* **A User creates a Workspace.**
* **A User joins a Workspace as a Workspace Member.**
* **A Workspace has multiple Projects.**
* **A Project contains multiple Tasks.**
* **A User is assigned a Role in a Workspace.**
* **A Task is assigned to a User.**
* **A Workspace has multiple Members with different Roles.**
* **A User can have multiple authentication Accounts (Google Email, etc.).**

**Slide 12: User Auth Flow**

**Divya:**  
Now, let’s walk through how the different process takes place in the system. Let, break it down in different steps. The first one is user authentication flow.

**Ghanshyam:**If user don’t have account, user can register, otherwise user can simply log in.  
When a user logs in, the system checks if they already have an account and workspace.

**Slide 13: Workspace handling**

**Ghanshyam:**If not, a **default workspace (“My Workspace”) is automatically created**. **Especially when new user shows up and user is assigned with the owner role of its workspace.**  
If they already have one, the system redirects them to their first available workspace.

**Slide 14: Workspace Management**

**Divya:**  
The **dashboard** is where users can **create and manage their workspaces**. Once a workspace is created, **Owners and Admins** can **invite new members**, adding them to the **Workspace Members list**.   
They also have the **power to change roles**, ensuring the right people have the right level of access.   
And if needed, **Owners and Admins** can also **remove members** from the workspace.

**Slide 15: Project Management**

**Ghanshyam:**Just like the workspace management, users with roles **Admin and Owner** can **create a project** within a workspace. The Project can be managed by the 2 roles, that is can be **edited and deleted** by the **Admin** and **Owner**.

**Slide 16: Task Management**

**Divya:**  
From the **Project Dashboard**, **Tasks can be created**. **Any role** can **create the task** that is **Owner, Admin and Member**. The task can be **assigned** to **any** **user** by **anyone**.   
**Users with any role** is able to **edit the tasks.** But only **Admin and Owner can delete** the **task**. Each time a **task is created, edited or deleted** the **Task** **Collection** will change accordingly.

**Ghanshyam:**  
And if you are wondering why we used these parts of the flow charts to demonstrate the flow, instead of the whole flow chart at once

**Slide 17: Summary, flow chart**

**Ghanshyam:**  
This is what it looks like. Anyways, so in a nutshell, in this application, the flow is something like:

* **Default Workspace ("My Workspace")** is created automatically if the user has no workspace**.**
* Only **Owners** and **Admins** can manage workspace members and roles.
* **Members Role** cannot **create, edit, or delete Projects and workspaces.** They can only **create** and **edit Tasks**
* **Owners, Admins,** and **Members** can **create and edit tasks,** but only **Owners and Admins** can delete them**.**

**Slide 18: T-ming UI-auth page**

**Divya:**  
Here we have some images of our application’s UI,  
This is the first page or the Registration and Login page of the application.

**Slide 19: T-ming UI-workspace dashboard**

**Ghanshyam:**  
This is the workspace dashboard. We can see the total number of tasks, Overdue and completed tasks. Also, we can see the Recent Projects, recent tasks, and recent members of our workspace in the bottom section right here if you can see.

**Slide 20: T-ming UI-Manage workspace**

**Divya:**  
These 2 images are about the workspace management, this one is the create workspace dialog, and this one is update and delete workspace page, which is just the workspace setting page.

**Slide 21: T-ming UI-Create and Edit Project**

**Ghanshyam:**  
Here these are 2 dialog boxes, first one shows the create project dialog, and other one is edit project dialog, so that you can change the different values of the project.

**Slide 22: T-ming UI-Project Dashboard**

**Divya:**  
This particular section is the Project dashboard, the section where we can see the different tasks within a project with an analytical dashboard of the project itself within. From here we can edit the project itself, and also create a task within the project itself.

**Slide 23: T-ming UI-Task Create and edit dialog**

**Ghanshyam:**So, the create task dialog, the first one, when you click on the create task will appear, and you can create the task, and the other one, this is the edit task dialog, when you need to change some task value such as its status or something else, this dialog will show up.

**Slide 24: T-ming UI-Tasks list**

**Divya:**  
Here are the task, how they will appear in as a tabulated list. The tasks can be sorted based on different values such as based on priorities, status etc.

**Slide 25: T-ming UI-Manage workspace**

**Ghanshyam:**  
Here, the first image shows that we can filter out the tasks, that is, we can see only task with a particular value of some particular field within the task. And the other one shows the assigning role to the member in the member dashboard.

**Slide 26: T-ming UI-Manage workspace**

**Divya:**  
This application aims to provide a **simple, efficient, and scalable solution** for managing team**-based or solo projects, or tasks,** whileensuring **secure authentication** and **access control.** Using modern technology, it provides a **simple and clear way** for teams to **work together without confusion**. With **automated workspace setup and clear permissions**, teams can **focus on their work more efficiently**, making project management **more organized and user-friendly**.

**Ghanshyam:**  
That’s **T-ming**—an application to manage your projects. Crafted with code and computers by us, Ghanshyam

**Divya:**And Divya

**Ghanshyam:**  
Thank you for listening! We’d love to hear your thoughts and answer any questions.