Developing a Full-Stack Application Using SAP Build Code and SAP Joule

By Vishal Gundage, Senior Consultant (Service Delivery) at On Device Solutions

Tutorial Overview

This tutorial describes how to build an Employee Expense Management Application using SAP Build Code, CAP (Cloud Application Programming Model), and Joule AI.

The application demonstrates how Generative AI and low-code tooling can accelerate enterprise application development while maintaining strong governance and extensibility.

You will create backend data models and services, enhance sample data, implement business logic for expenses and approvals, and finally generate Fiori-based user interfaces.

Prerequisites:

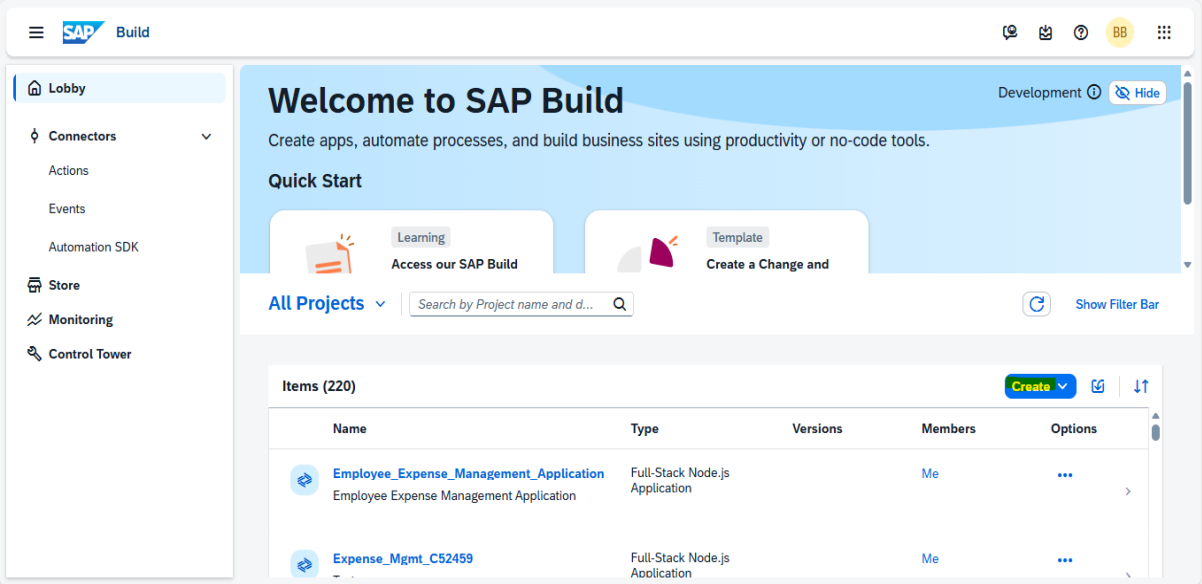
- Access to **SAP Build Lobby**
- An active **SAP Business Application Studio Dev Space**
- Basic understanding of CAP and SAP Build Code concepts

Creating Backend Logic for Employee Expense Management Application

This section focuses on the development of backend logic using SAP Build Code and Joule AI Copilot. It covers the creation of data entities, backend logic, and integration with SAP systems.

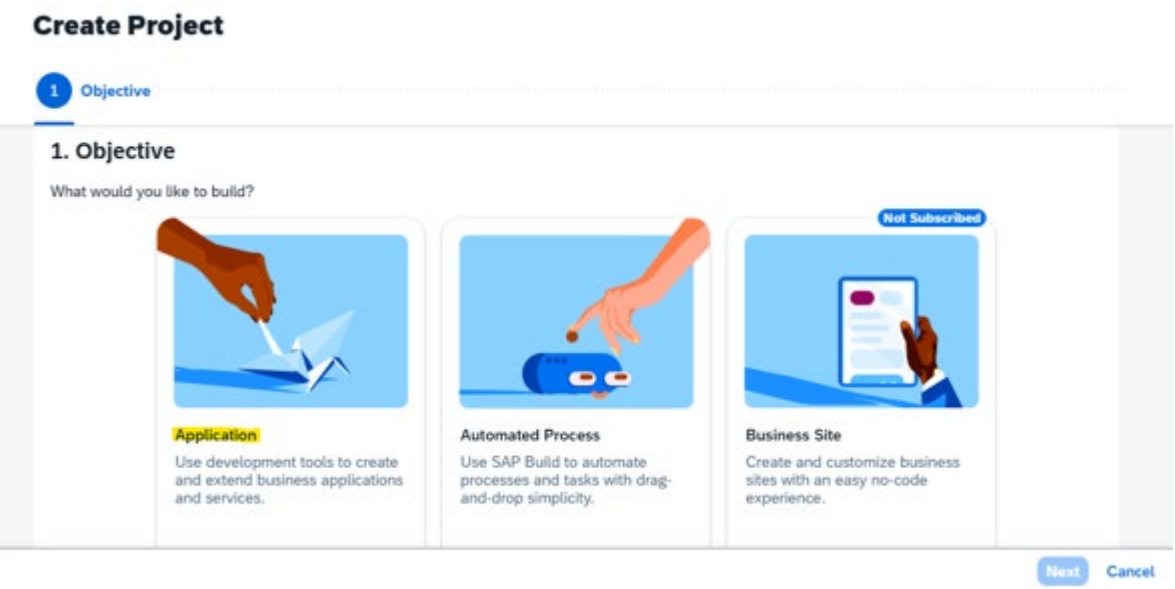
Launch SAP Build Code and Create Application

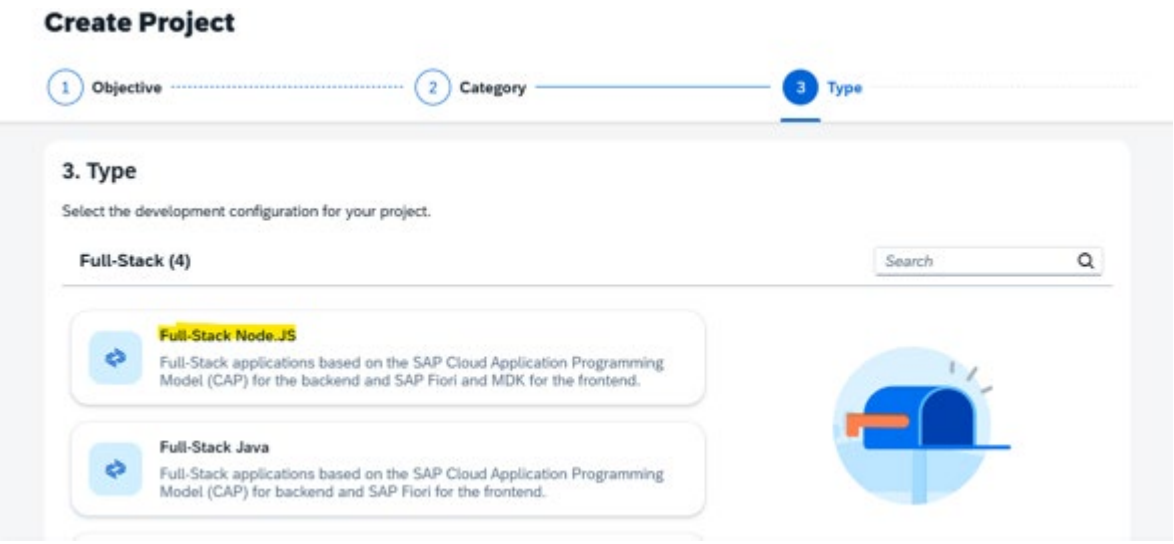
1. Log in to **SAP Build Lobby**
2. Click on **Create**
3. Select **Build an application**



Select Application Category and Application Type

1. Select Application.
2. Click Next.
3. Select Full-Stack.
4. Select Full-Stack Node.js.
5. Click Next.





Enter Project Details

Fill in the following details:

- Project Name:

  Expense_Mgmt_C52459 **or** Employee_Expense_Management_Application

- Description:
  Employee Expense Management using SAP Build Code and Joule AI

- Dev Space:

  Existing or Create New

Click **Review**

Application created and added to the Lobby.

Once the application is created, click on it to open the Storyboard.

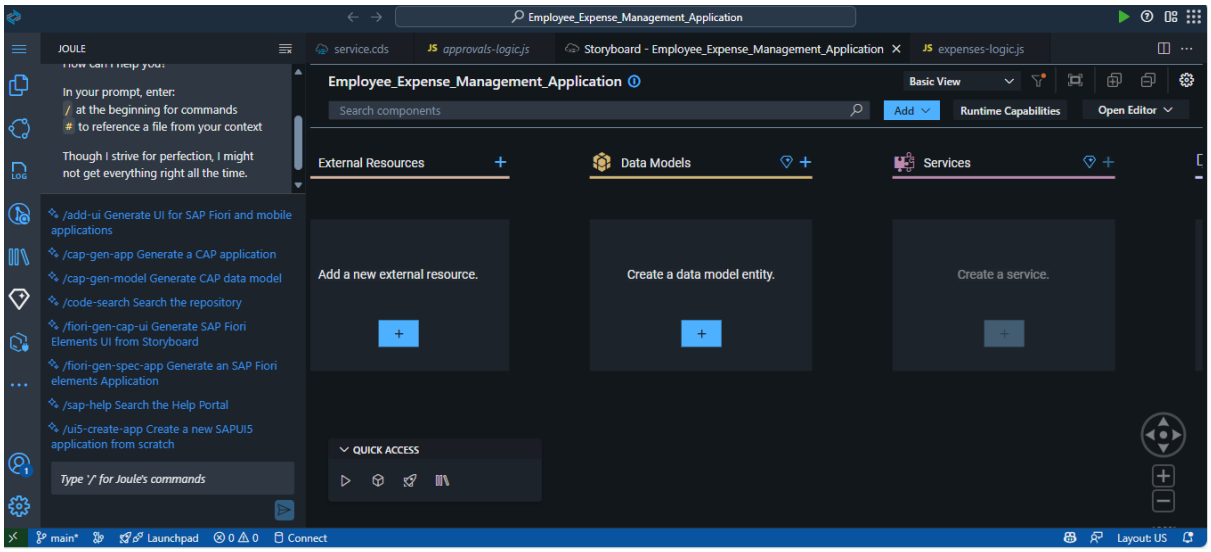
SAP Build Code will launch in the background, utilising the SAP Business Application Studio.

Please be patient while it opens….

Open Storyboard

1. Click on the newly created application.
2. Open the Storyboard.
3. Confirm cookie settings if prompted.

SAP Build Code launches SAP Business Application Studio in the background.



Create Data Entities with Joule AI

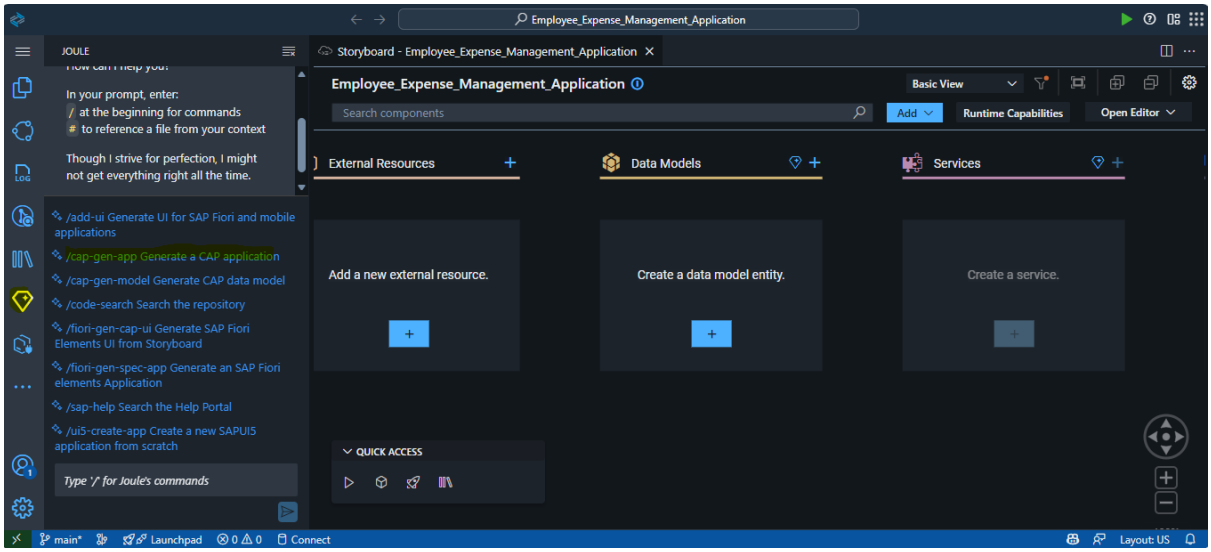
Launch SAP Joule AI Digital Assistant

1. Click on the SAP Joule icon.
2. Select command:

   /cap-gen-app – Generate a CAP application

What this step does -- /cap-gen-app

- Launches SAP Joule, the GenAI digital assistant embedded in SAP Build Code.
- The command /cap-gen-app instructs Joule to generate a Cloud Application Programming (CAP) project.
- Automatically creates data models, services, and project structure based on the provided prompt.
- Accelerates backend development by scaffolding CDS entities, service definitions, and basic configurations.
- Enables vibe coding, where developers describe intent while Joule handles repetitive setup tasks.



Joule Prompt – Data Model and Services

Paste the following prompt into Joule:

Design an employee expense management application.

Define 3 data entities: Employees, Expenses, and Approvals.

Employees must have the following fields:

- name (string)

- email (string)

- employeeId (7-digit integer)

- department (string)

- grade (integer)

- availableBudget (integer)

Expenses must have the following fields:

- expenseAmount (integer)

- expenseType (string)

- expenseDate (date)

- status (string)

- totalAmount (integer)

Approvals must have the following fields:

- approvalLevel (integer)

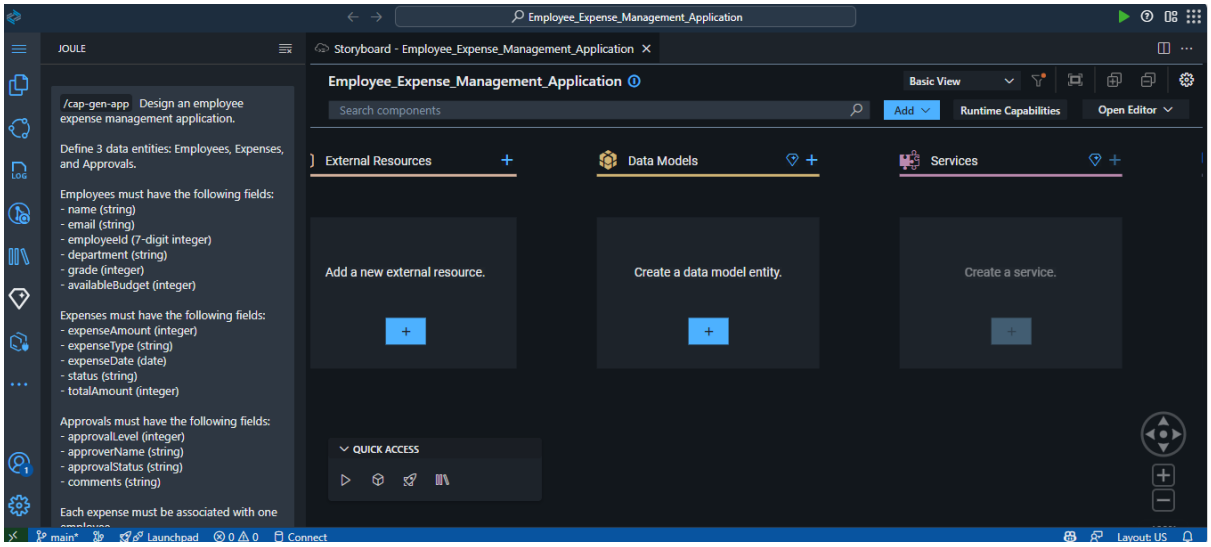
- approverName (string)

- approvalStatus (string) - comments (string)

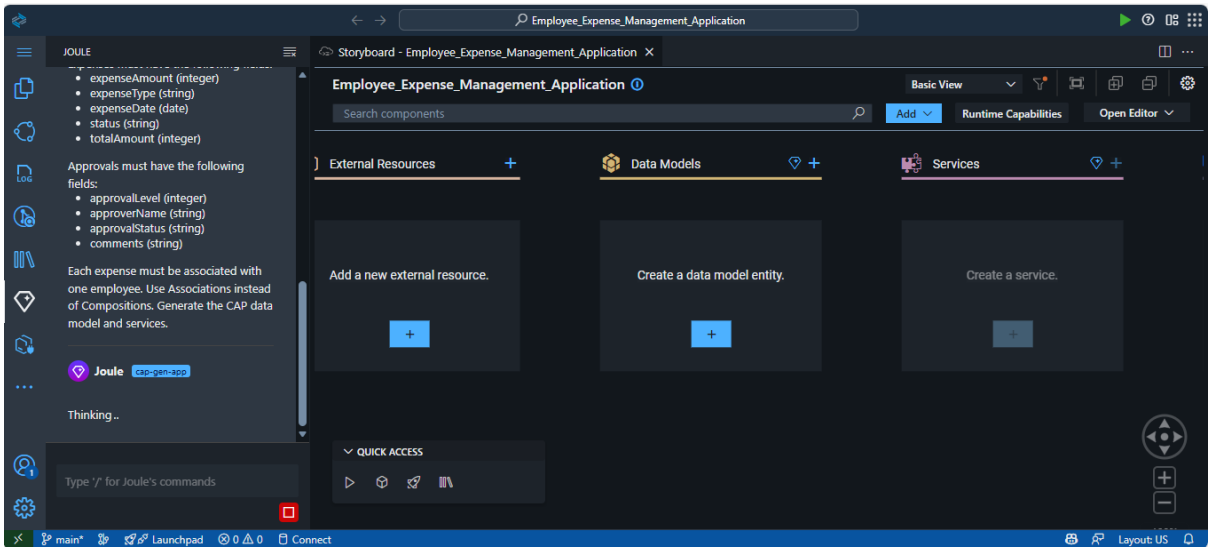
Each expense must be associated with one employee.

Use Associations instead of Compositions.

Generate the CAP data model and service.

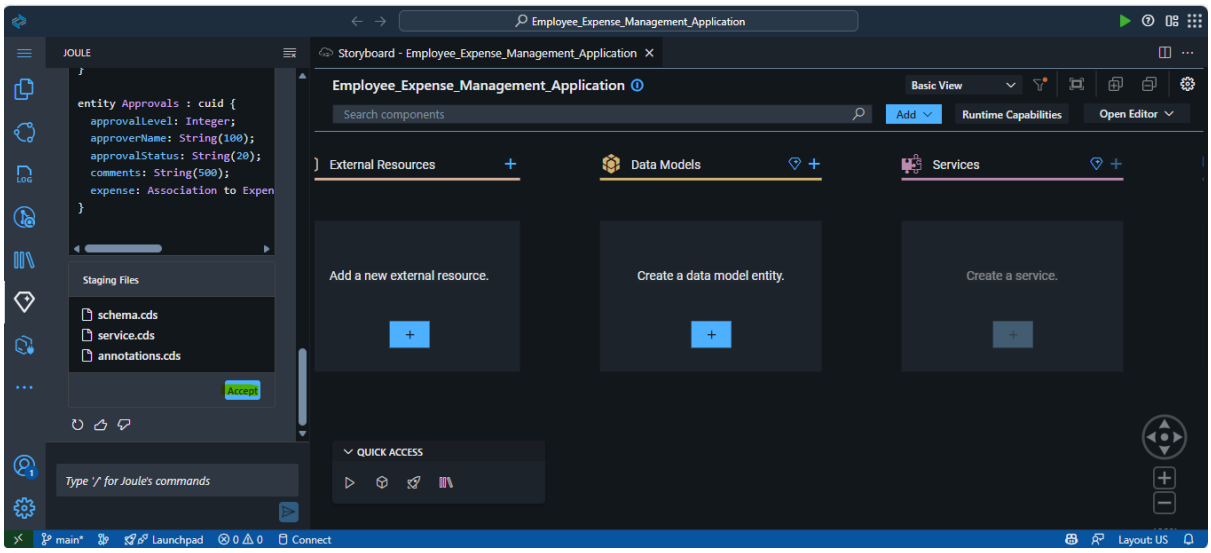


Joule Thinking...

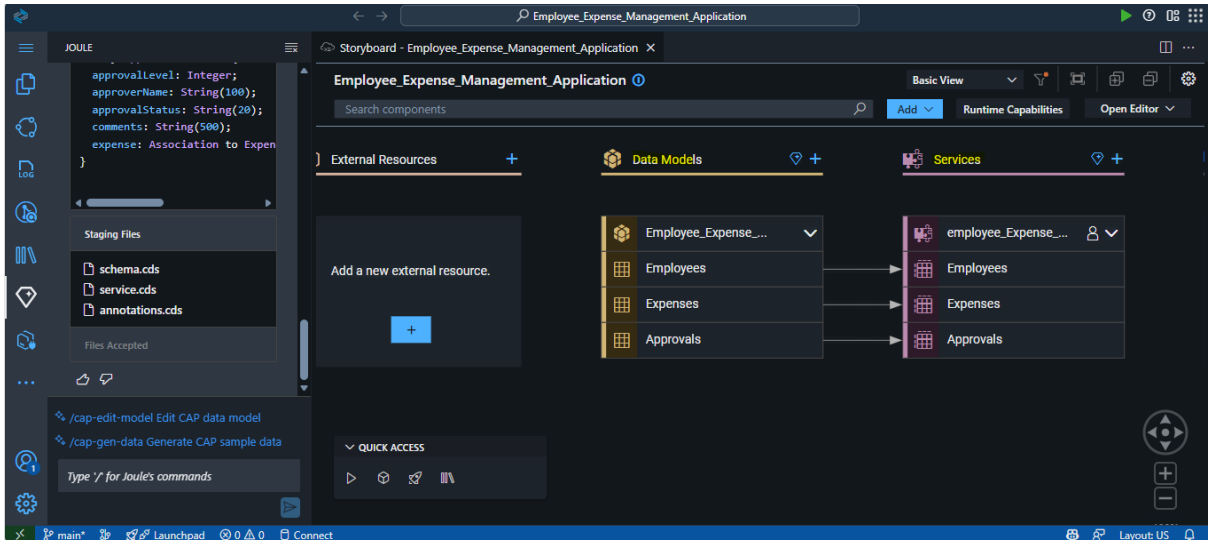


Review the generated CDS and service code.

Click **Accept**.



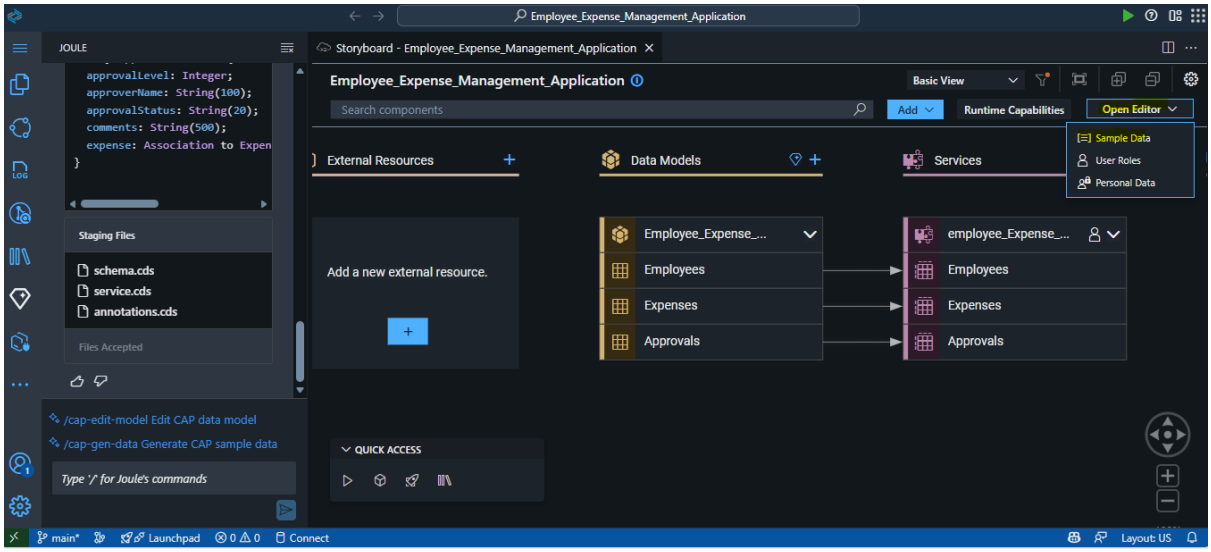
Data Models and Services appear in the Storyboard.



Enhance Sample Data with Joule

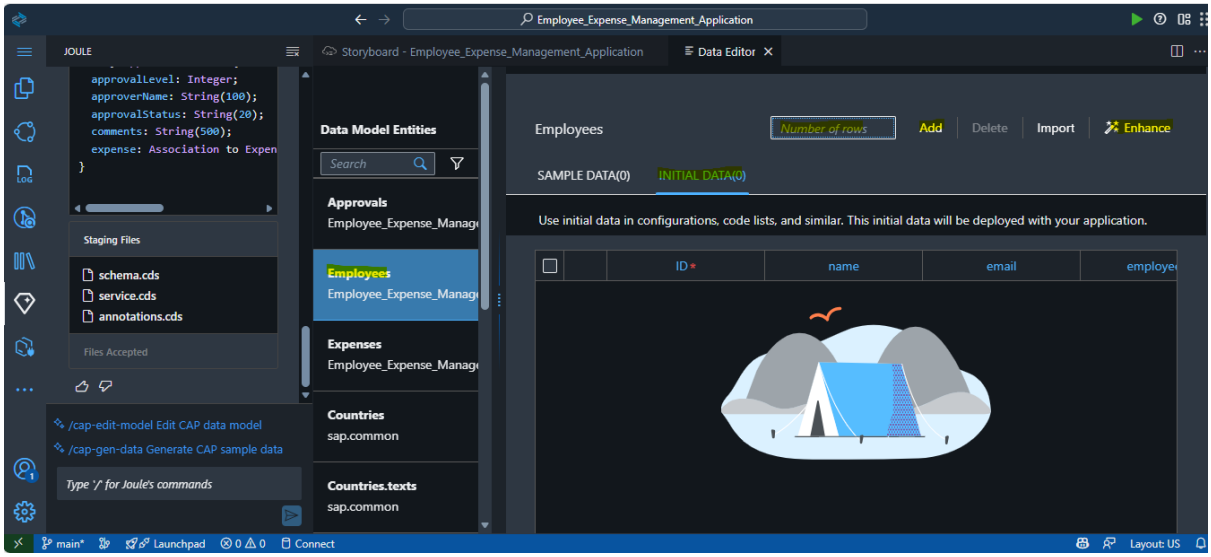
Open Sample Data Editor:

1. In Storyboard, click Open Editor.
2. Click Sample Data.



Add Sample Data – Employees

1. Select the Employees entity.
2. Select the INITIAL DATA tab.
3. Enter 5 rows.
4. Click Add.
5. Click Enhance.



Joule AI Prompt – Employees Sample Data

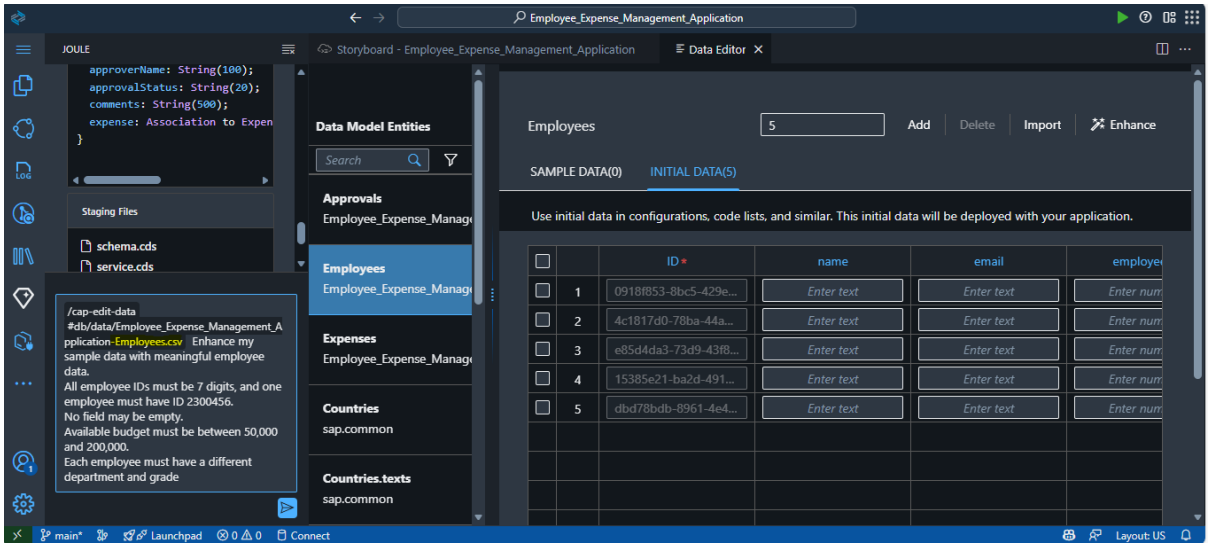
“Enhance my sample data with meaningful employee data.

All employee IDs must be 7 digits, and one employee must have ID 2300456.

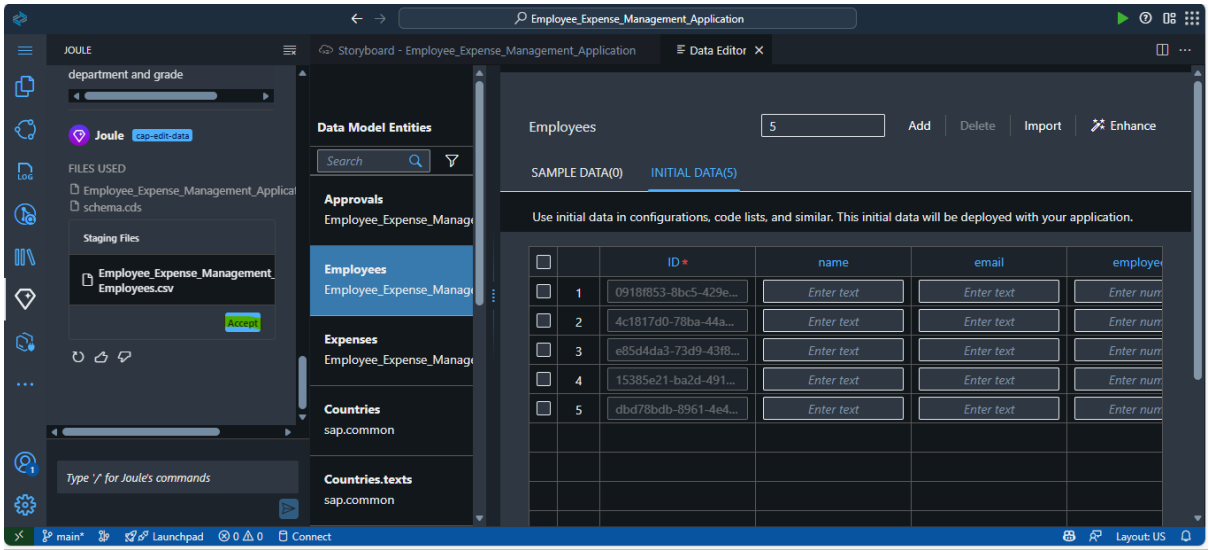
No field may be empty.

The available budget must be between 50,000 and 200,000.

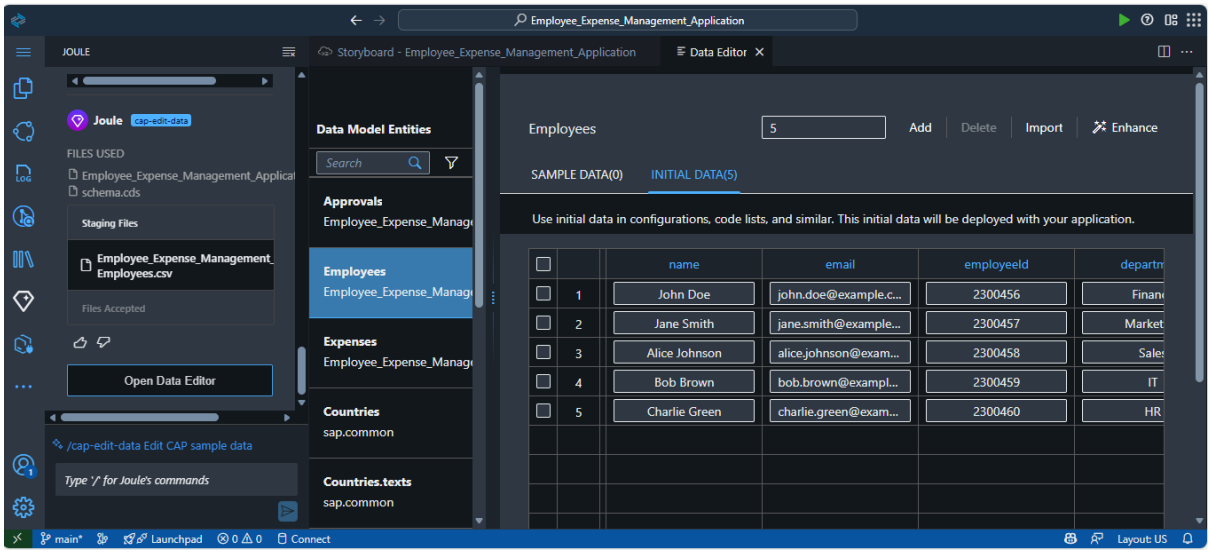
Each employee must have a different department and grade.”



1. Click **Generate**.
2. Review the data.
3. Click **Accept**.

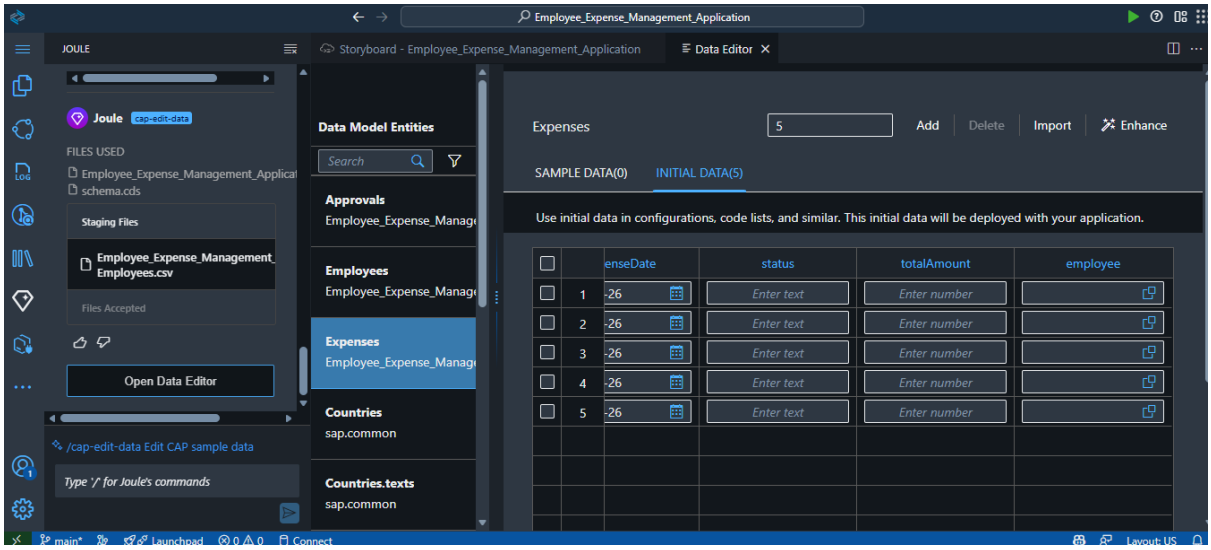


Employee sample data added.

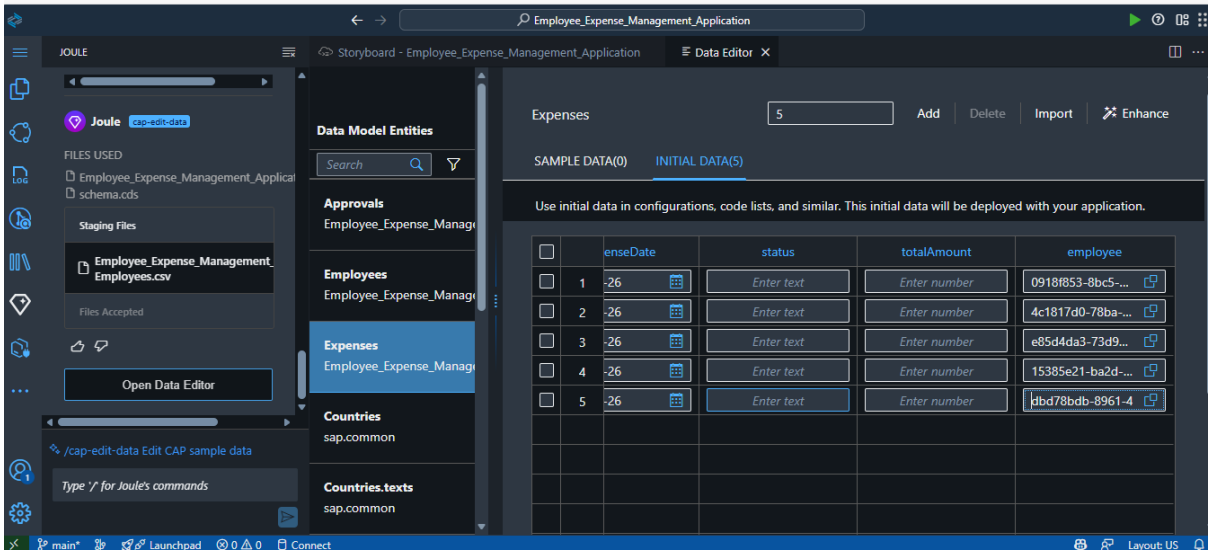


Add Sample Data – Expenses

1. Select **Expenses entity**.
2. Select the **INITIAL DATA** tab.
3. Enter **5 rows**.
4. Click **Add**.
5. Assign a **valid Employee ID** to each record.
6. Click **Enhance.**



Assign a valid Employee ID to each record.



Joule AI Prompt – Expenses Sample Data:

“Enhance my sample data with meaningful expense records.

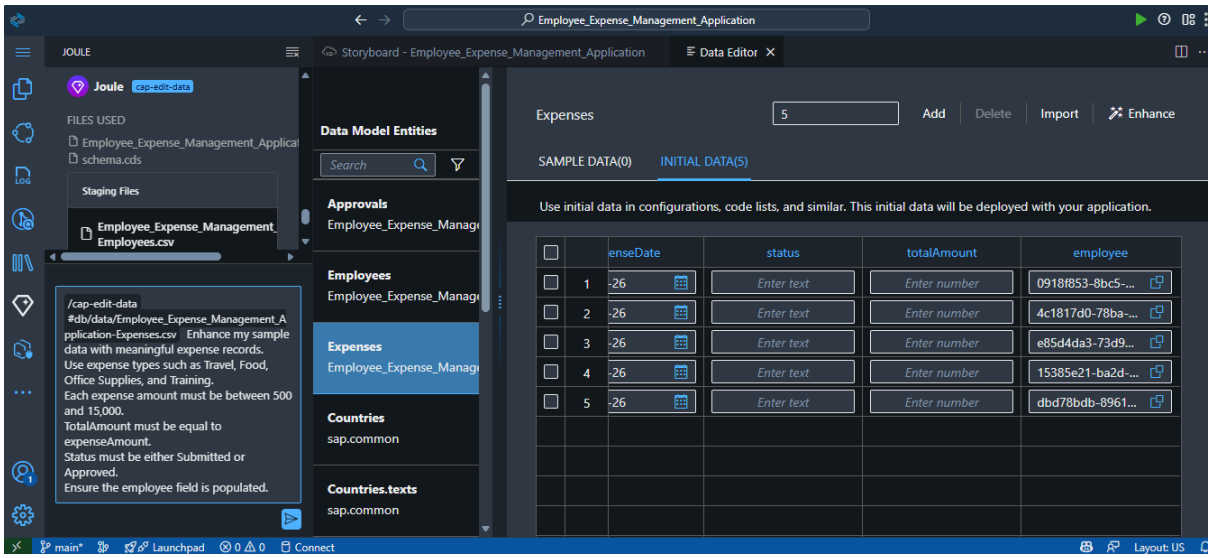
Use expense types such as Travel, Food, Office Supplies, and Training.

Each expense amount must be between 500 and 15,000.

TotalAmount must be equal to expenseAmount.

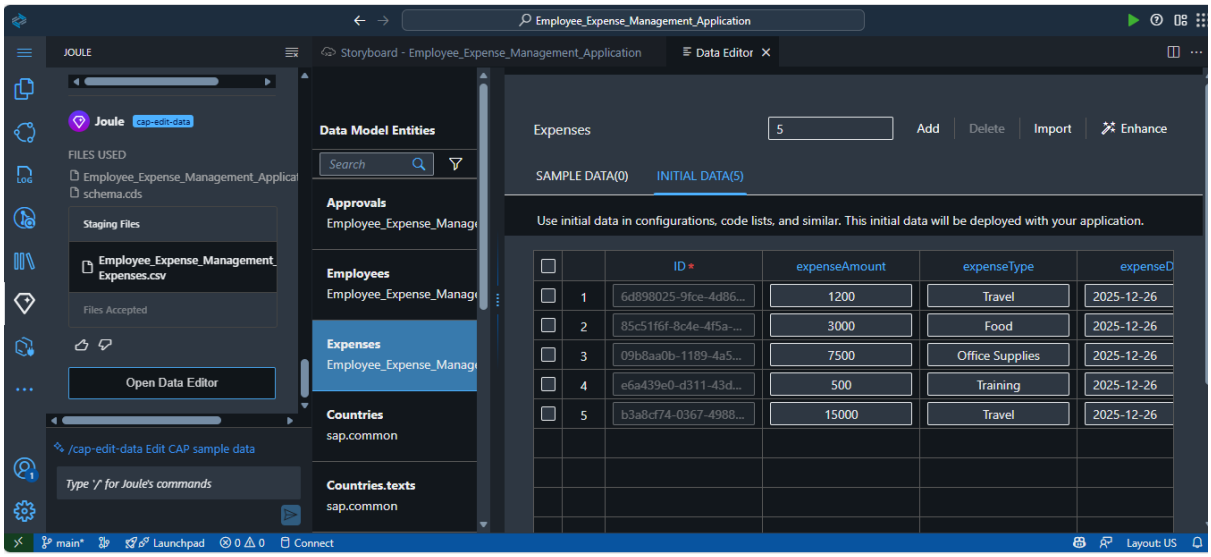
Status must be either Submitted or Approved.

Ensure the employee field is populated.”



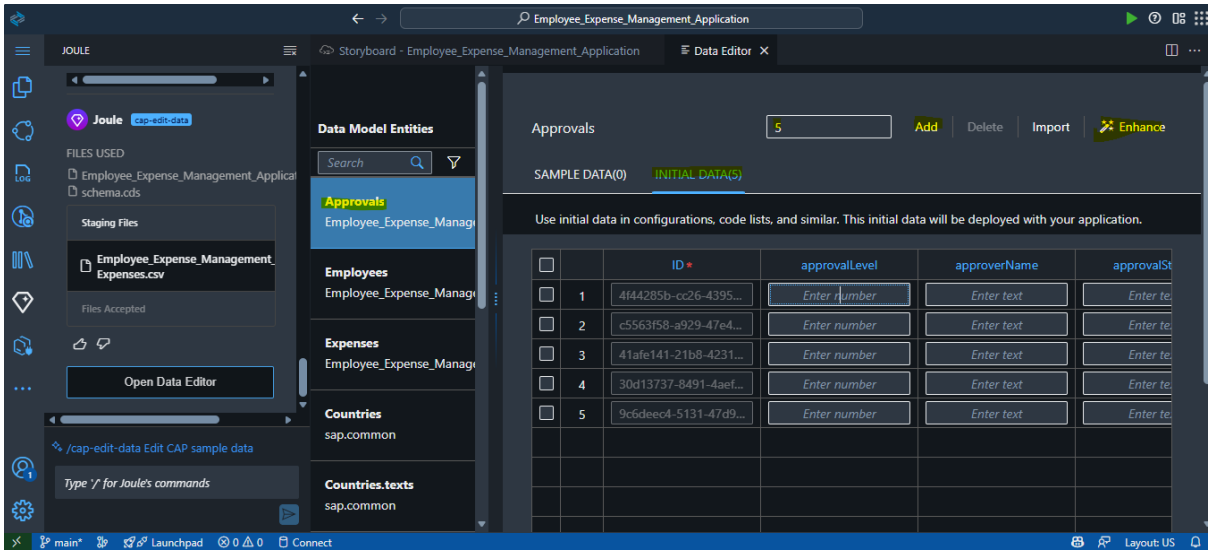
1. Click **Generate**.
2. Click **Accept**.

Expense sample data added.

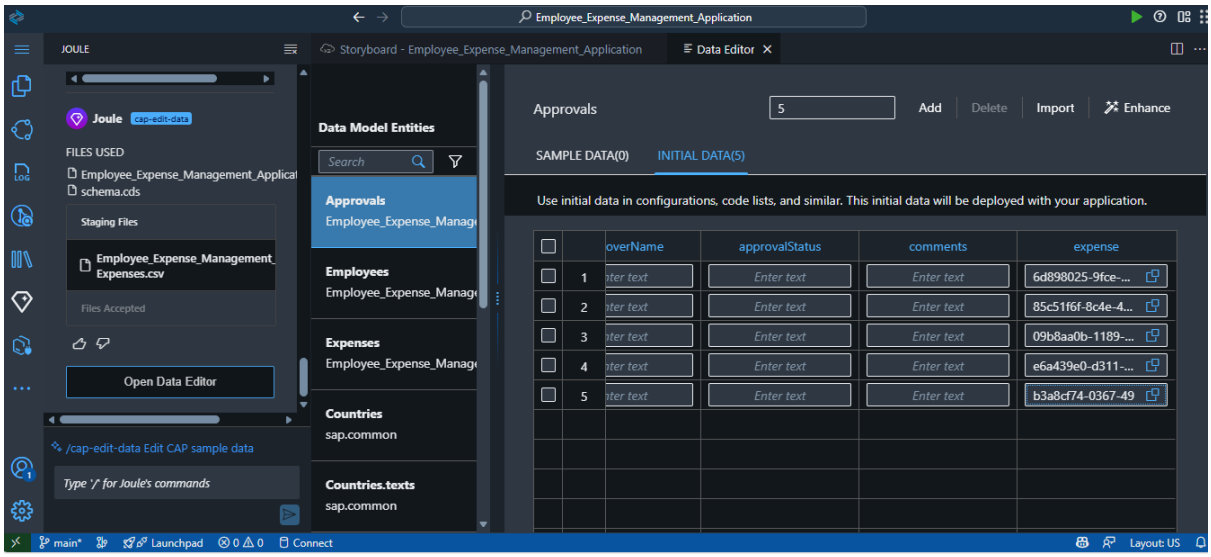


Add Sample Data – Approvals

1. Select **Approvals** entity.
2. Select the **INITIAL DATA** tab.
3. Enter **5** rows.
4. Click **Add**.
5. Assign a valid **Expense ID** to each record.
6. Click **Enhance**.



Assign a valid Expense ID to each record.



Joule AI Prompt – Approvals Sample Data:

“Enhance my sample data with meaningful approval records.

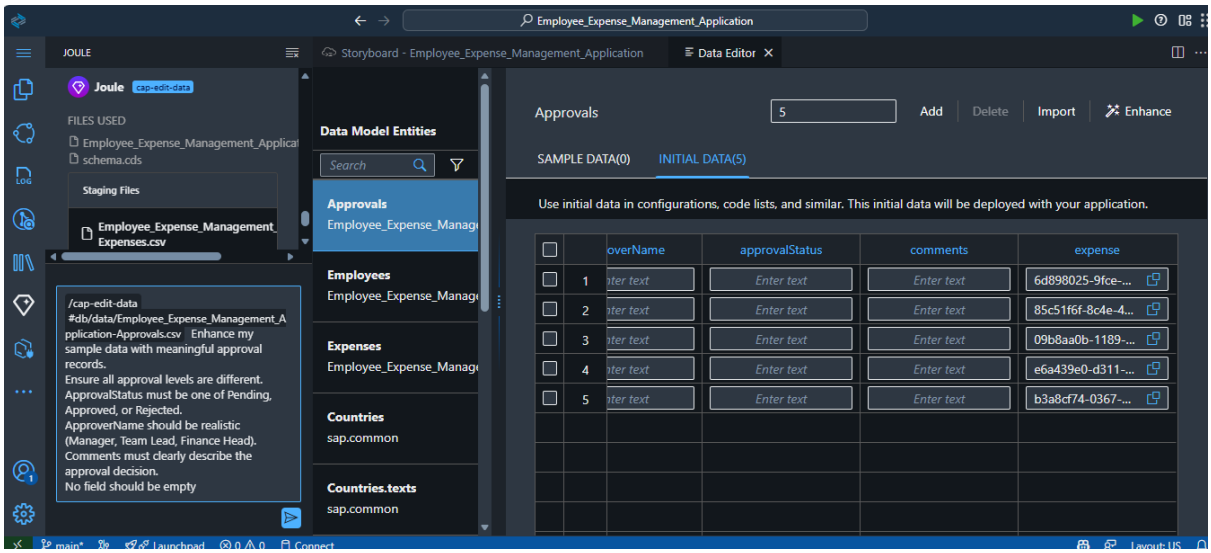
Ensure all approval levels are different.

ApprovalStatus must be one of Pending, Approved, or Rejected.

ApproverName should be realistic (Manager, Team Lead, Finance Head).

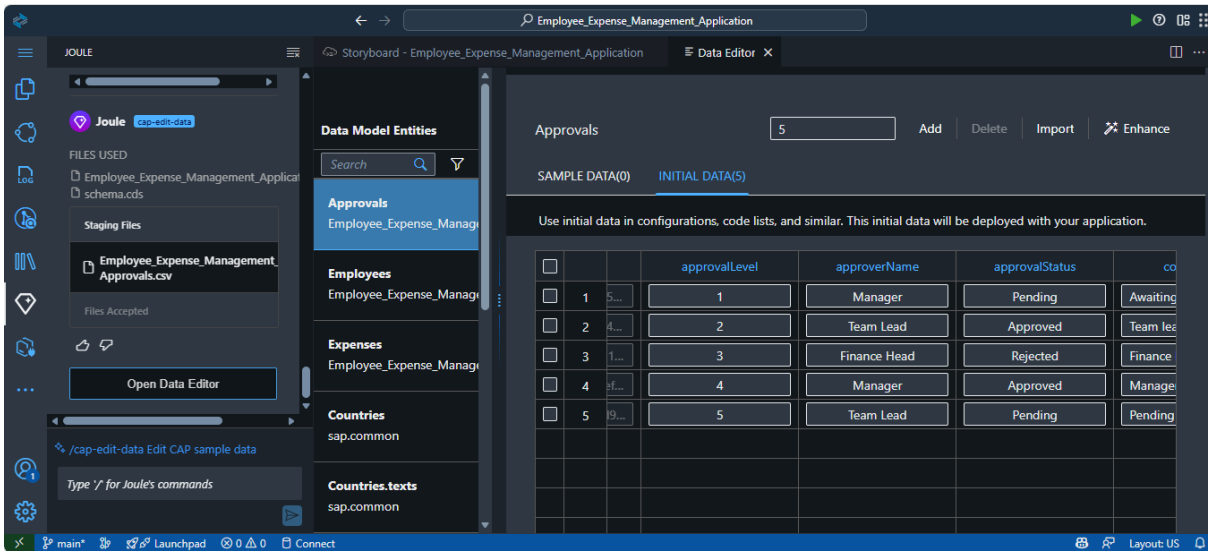
Comments must clearly describe the approval decision.

No field should be empty.”



1. Click **Generate**.
2. Click **Accept**.

Approval sample data added.



Create Backend Logic with Joule AI – Expenses

1. Go to Storyboard.
2. Under Services, select Expenses.
3. Click **Open in Graphical Modeler**

What is the Graphical Modeler in SAP Build Code?

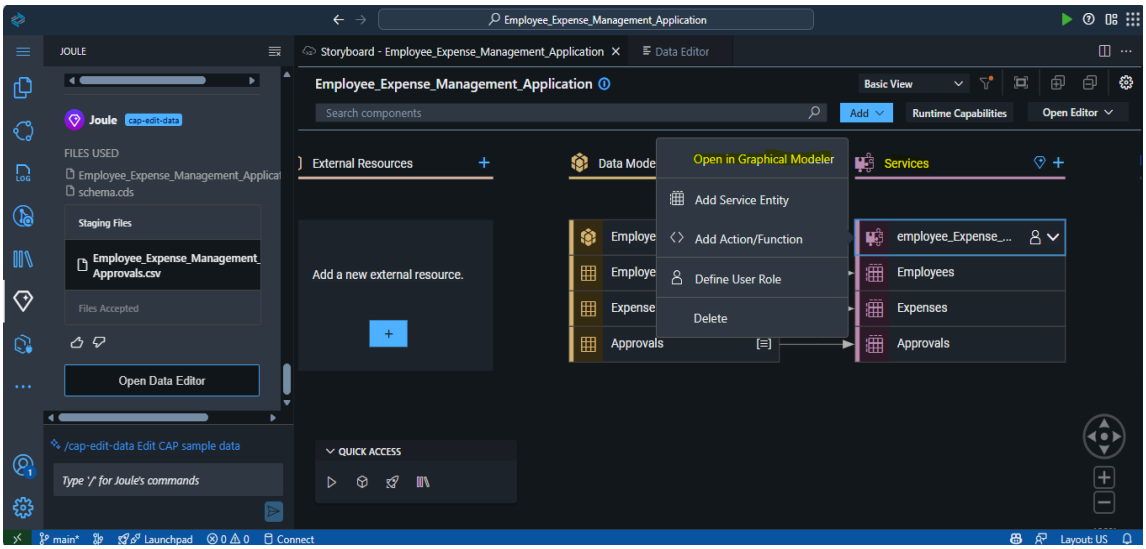
The Graphical Modeler in SAP Build Code is a visual design tool allowing developers to view, create, and enhance CAP data models and services using diagrams instead of only code.

What it is used for:

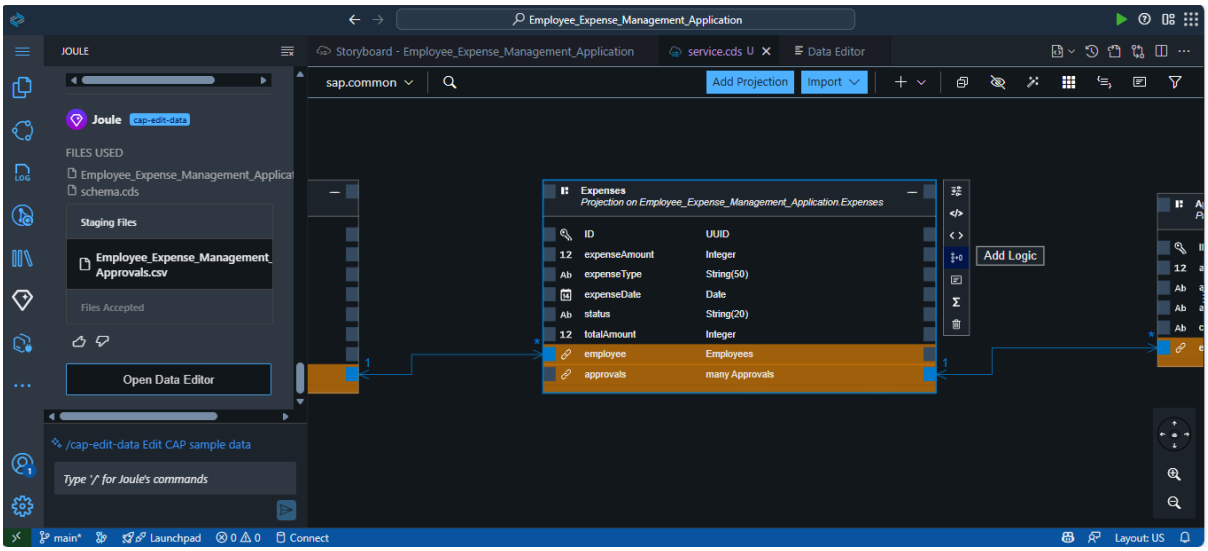
- Visually design entities, fields, and associations
- Add backend logic (Create, Update, save events) without digging into folders
- Quickly navigate from visual model generated CDS code
- Safely extend AI-generated models with human validation
- Understand relationships between Entities (Like Employees, Expenses, and Approvals) briefly

Why it matters (especially with Joule and vibe coding):

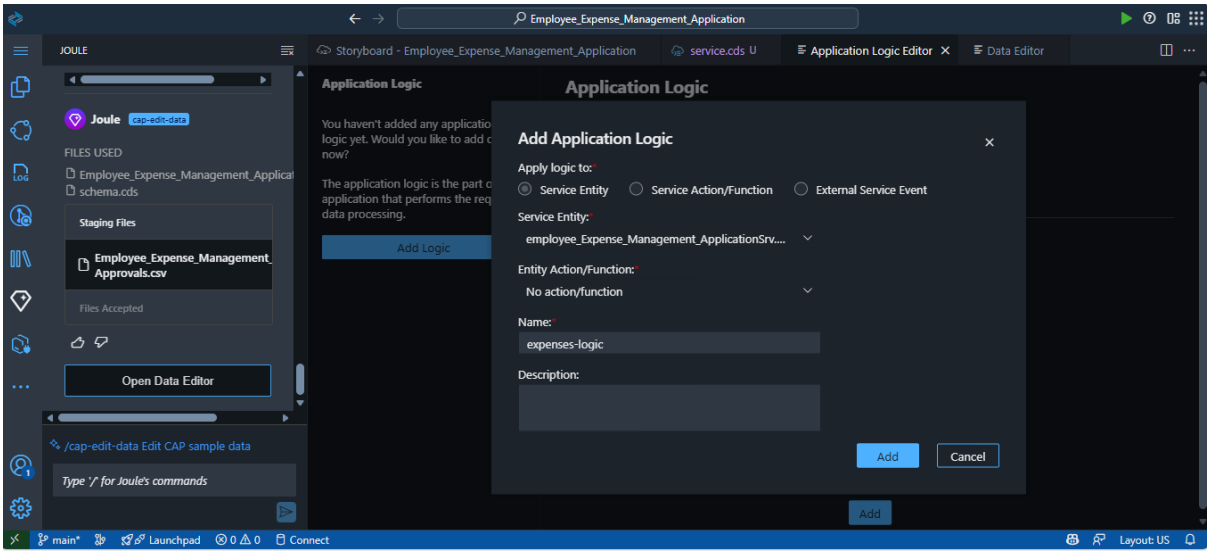
- Joule can generate the initial model
- Graphical Modeler helps review, correct, and refine it visually
- This keeps developers in control while benefiting from GenAI speed



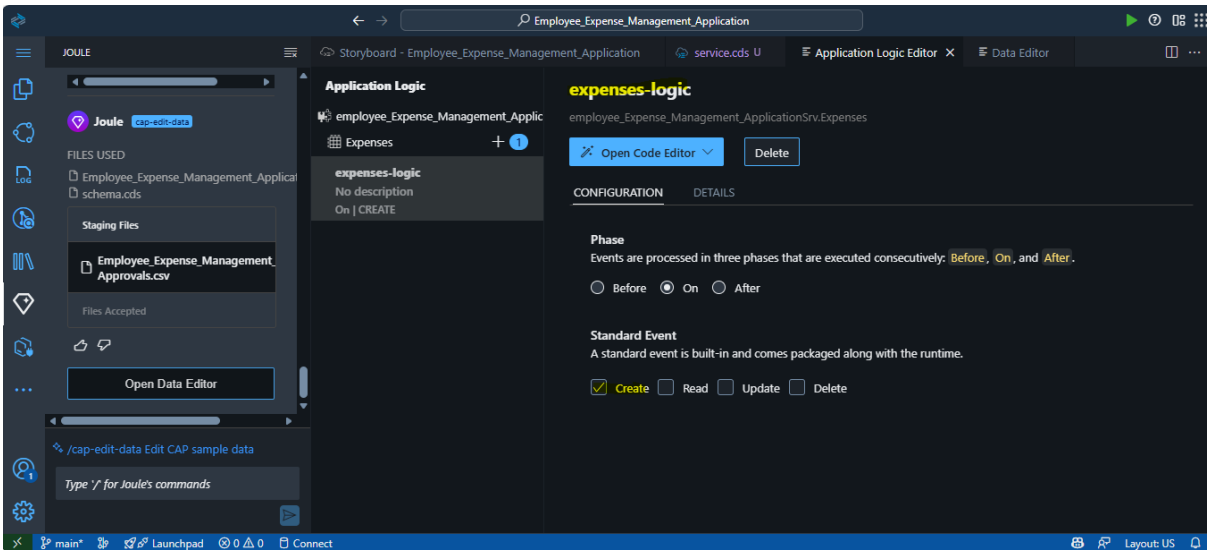
4. Select Expenses entity.
5. Click Add Logic.



6. Keep defaults → Click Add.

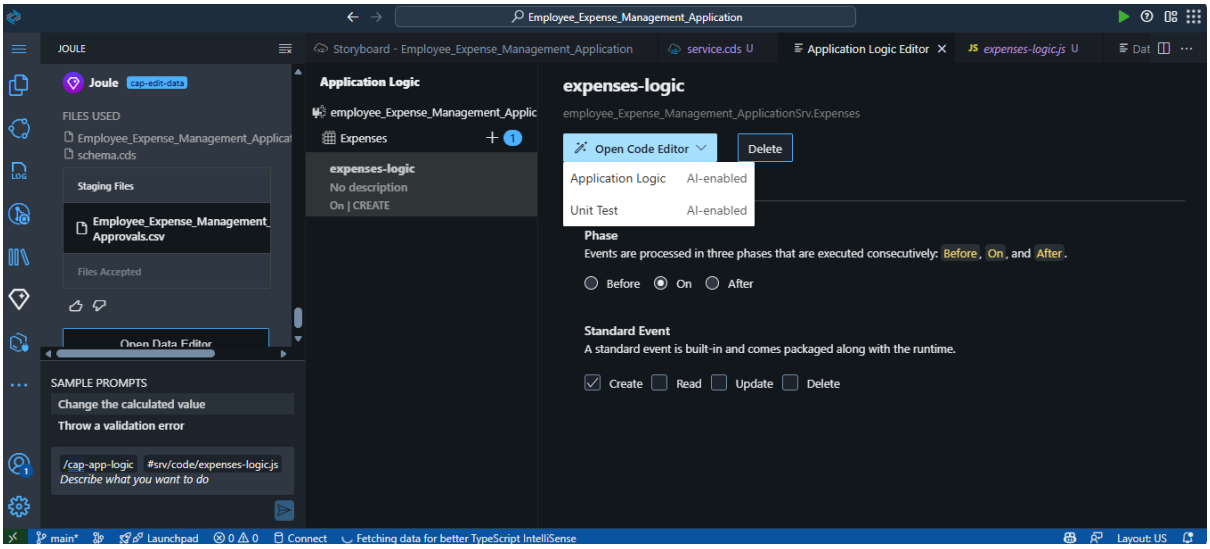


7. Event: Create



8. Click Open Code Editor

9. Select Application Logic AI



Joule AI Prompt – Expenses Backend Logic:

“Implement backend logic for the Expenses entity.

Before creating an expense:

- Fetch the related employee.

- If the employee does not exist, throw a business error.

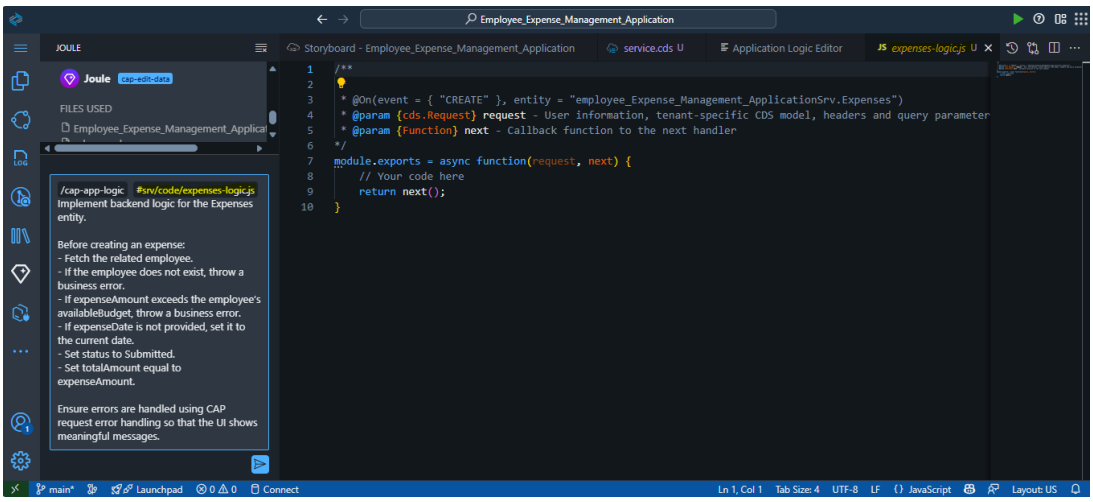
- If expenseAmount exceeds the employee’s availableBudget, throw a business error.

- If expenseDate is not provided, set it to the current date.

- Set status to Submitted.

- Set totalAmount equal to expenseAmount.

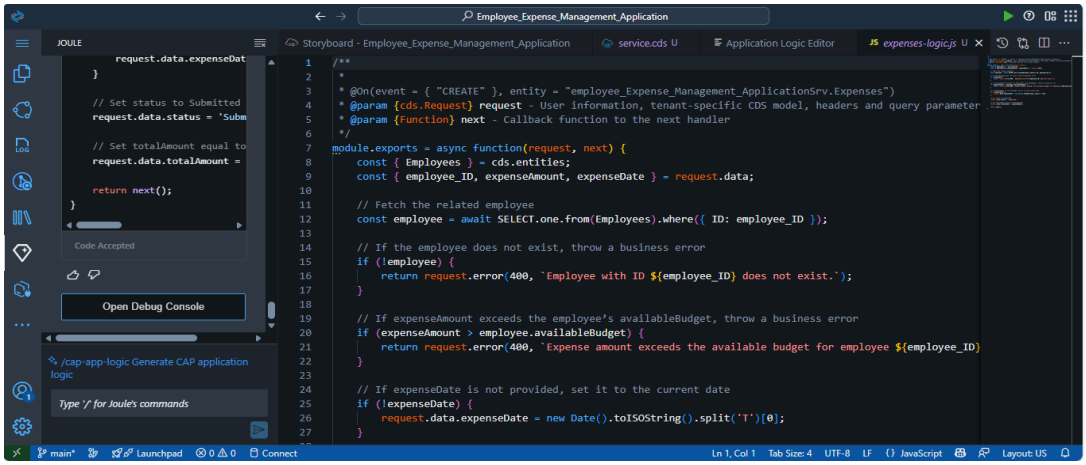
Ensure errors are handled using CAP request error handling so that the UI shows meaningful messages.”



Click **Generate**, review the code.

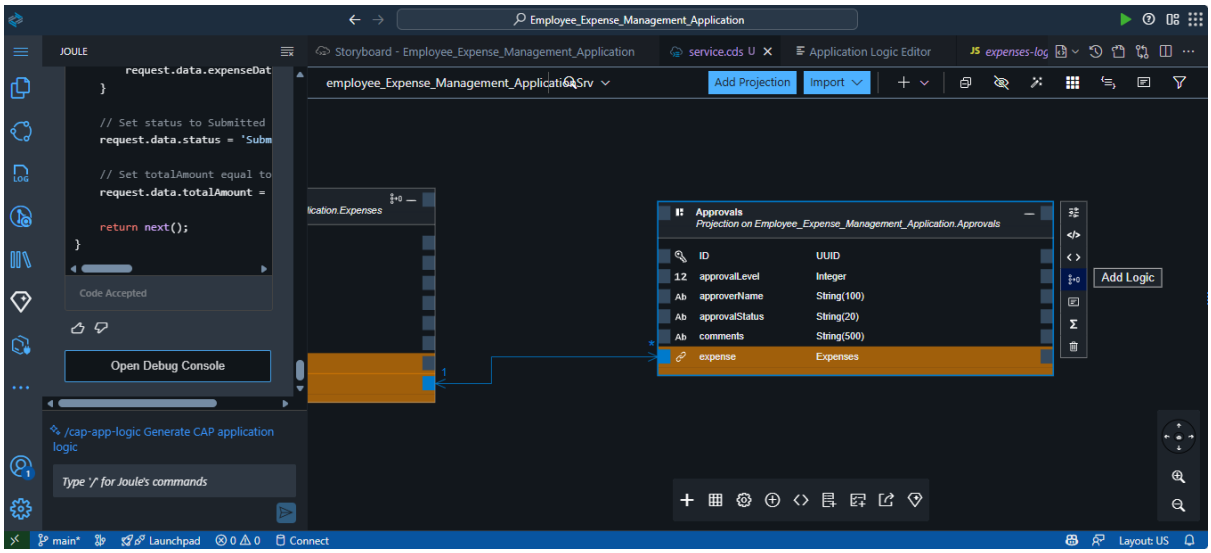
Click **Accept**.

Backend logic added for **Expenses**.



Create Backend Logic with Joule AI – Approvals

1. In **Storyboard**, select **Approvals** entity.
2. Click **Add Logic**.



3. Event: **Create.**
4. Click **Open Code Editor**.
5. Select **Application Logic**.

Joule AI Prompt – Approvals Backend Logic:

Implement backend logic for the Approvals entity.

- Validate that the related expense exists.
- Block approval creation if the expense is already approved or rejected.
- When approval status is Approved:
  - Validate the related employee exists.
  - Validate sufficient available budget.
  - Deduct the expense amount from the employee's budget.
  - Update expense status to Approved.
- When approval status is Rejected, update the expense status to Rejected.

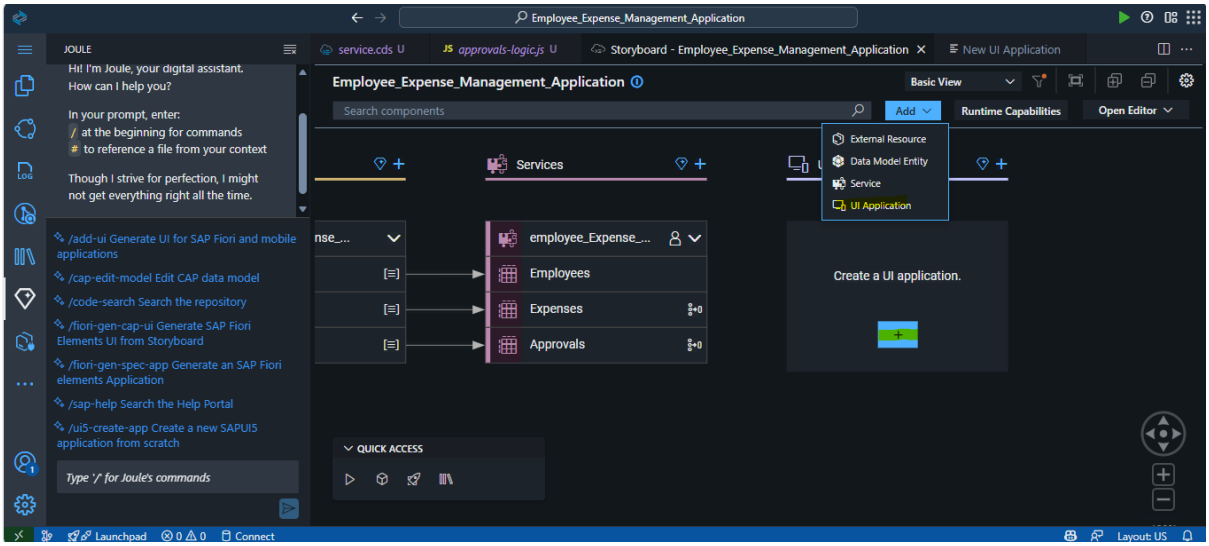
Raise meaningful business errors to prevent inconsistent data.

1. Click **Generate**.
2. Review the logic.
3. Click **Accept**.
4. Backend logic added for **Approvals.**

Add UI Applications from Storyboard

With the data model, sample data, and backend logic in place, we now proceed to create UI applications for each business entity.

1. Go back to the Storyboard.
2. Under UI Applications, click on the (+) button to add a new UI application**.**



Create UI for Employees

Employee UI Details

- **Display Name**:

  Employees

- **Description**:
  Manage Employees

- **UI Application Type**:

  Template-Based Responsive Application

- **UI Application Template**:

  List Report Page

- **Main Entity**:

  Employees

- **Integration Entity**:

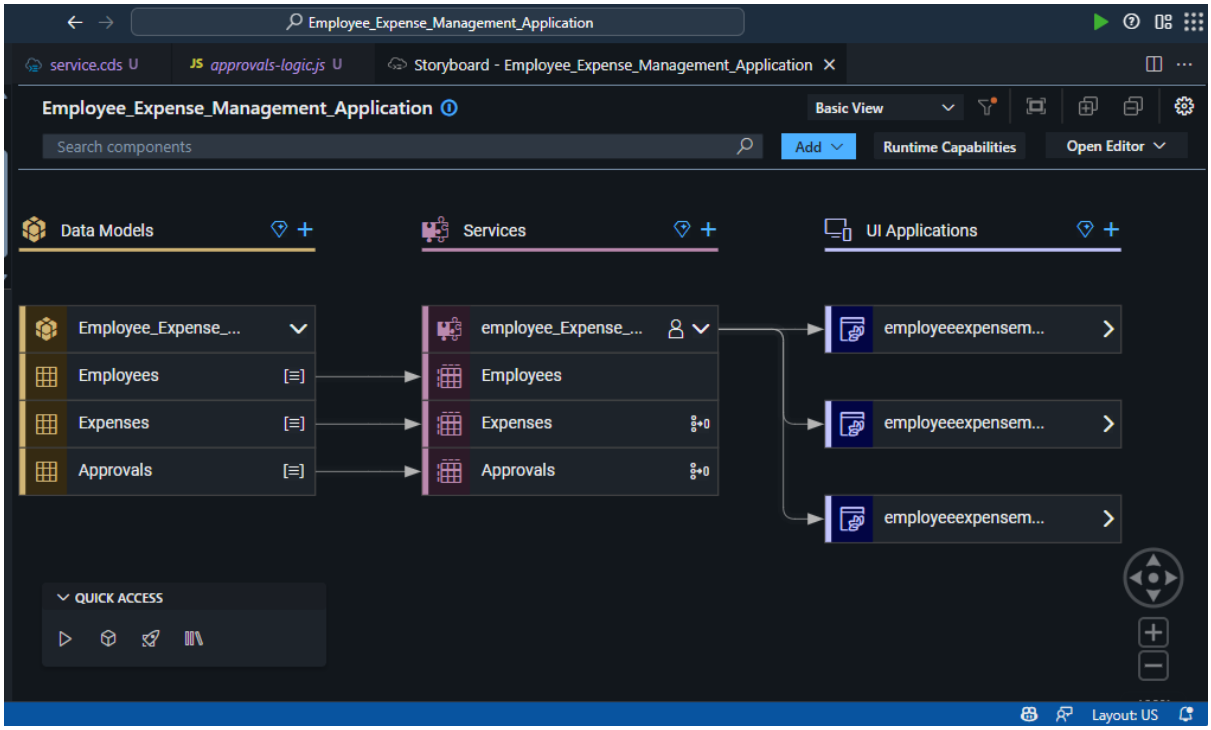
  None

Click **Finish**.

Take the same steps for creating the **UI for Expenses** and for **Approvals**

(Make sure to update the Main Entity and Display Name accordingly.)

Installing dependencies may take time – wait for it to finish.



Preview and Test the Application

Using SAP Build Code, we were able to generate enterprise-grade Fiori UIs for Employees, Expenses, and Approvals with just a few clicks. These UIs are fully connected to our CAP backend and enforce all business validations implemented earlier.

The application preview is displayed, showing the Fiori Application for all three entities: Employees, Expenses, and Approvals.

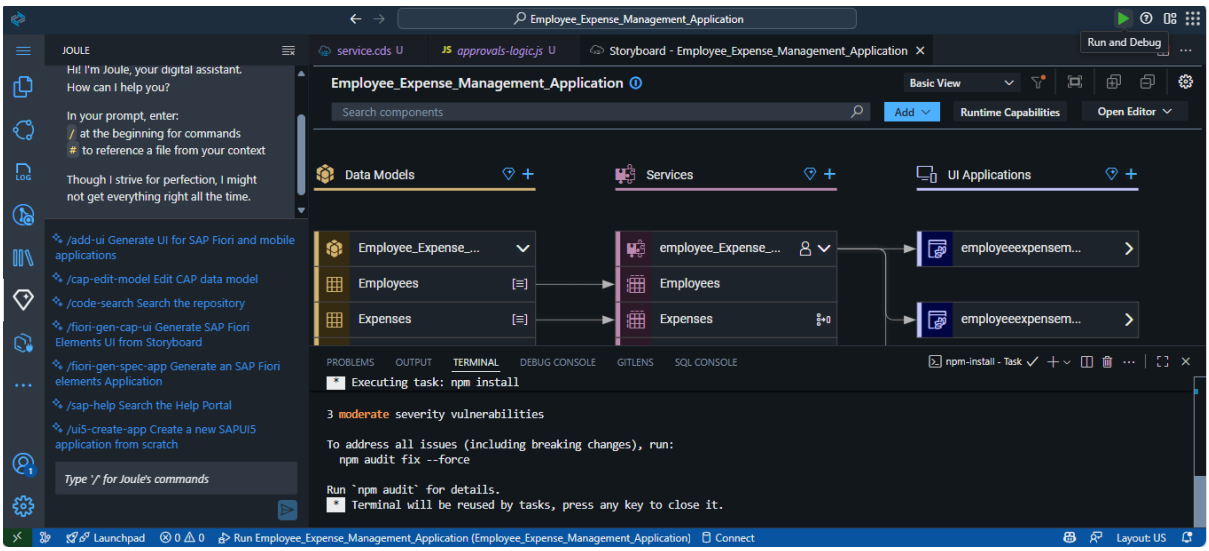
Click **Preview** from the toolbar.

Action 1 – Run the Application

The Application Developer navigates to the upper-right corner of the **SAP Build Code** interface and starts the application.

1. Click on **Run and Debug.**
2. Select the project:

   Employee_Expense_Management_Application
3. Click **Run.**

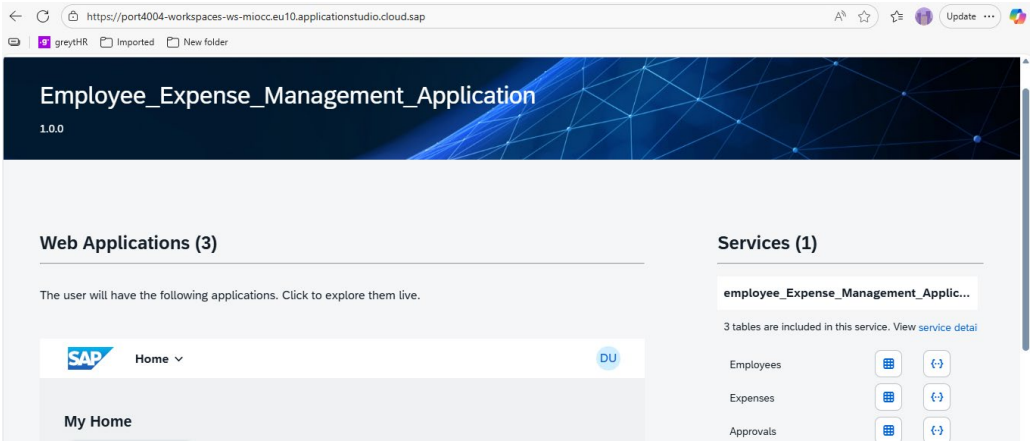


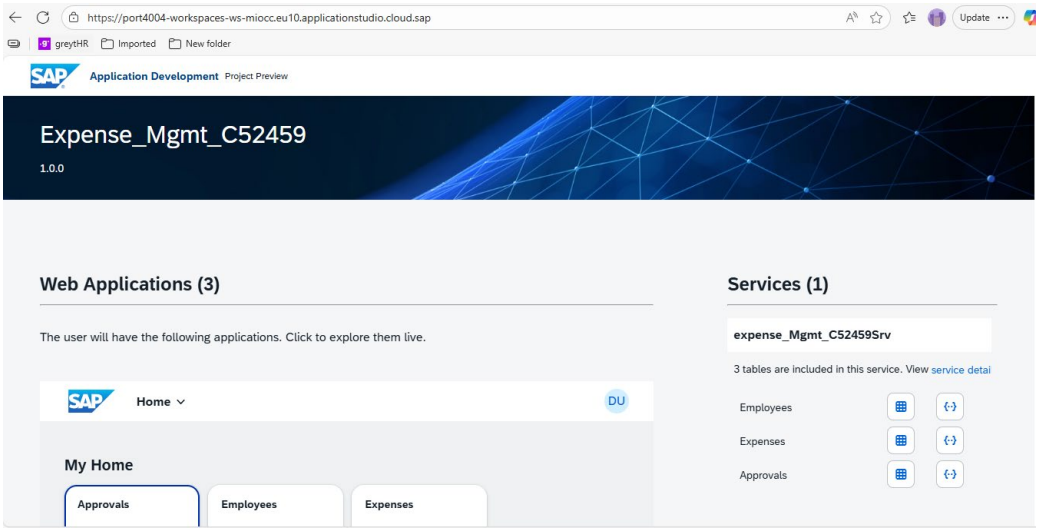
SAP Build Code builds and launches the application.

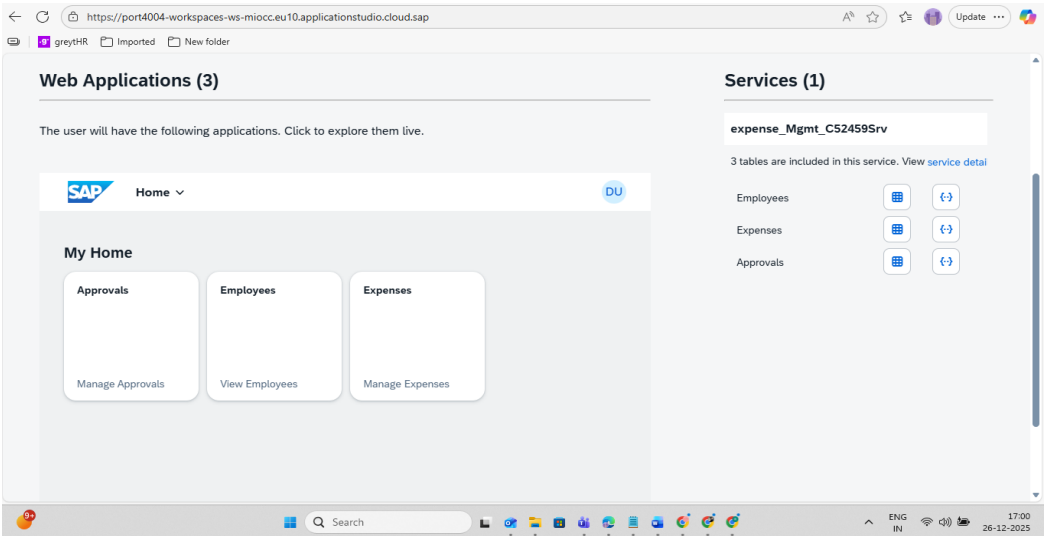
Display Application Preview

Once the application is running, the **Fiori Launchpad Preview** is displayed.

The preview shows **three tiles**: Employees, Expenses, Approvals.

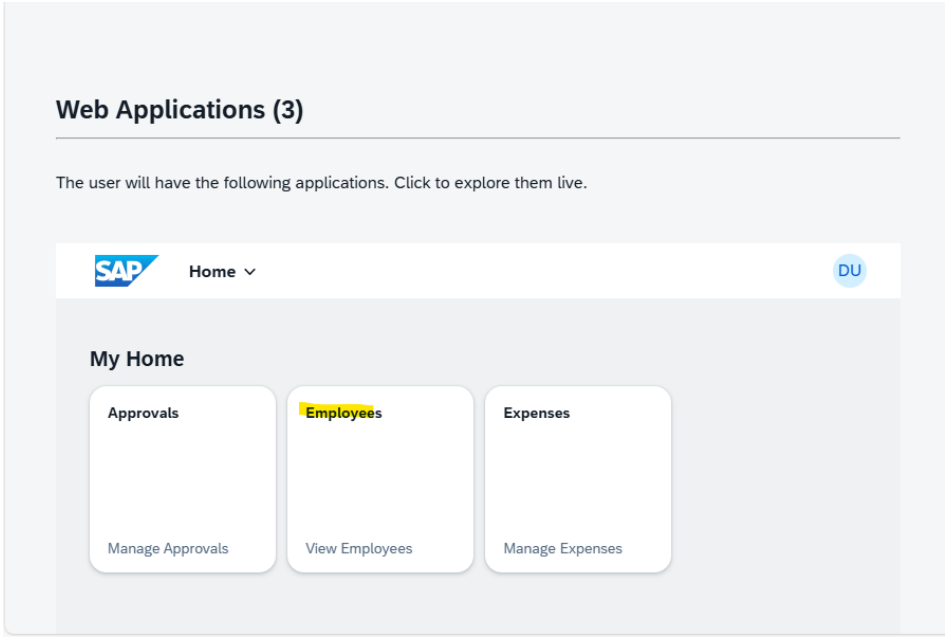

https://port4004-workspaces-ws-miocc.eu10.applicationstudio.cloud.sap
Update
greytHR
Imported
New folder
Employee_Expense_Management_Application
1.0.0
Web Applications (3)
The user will have the following applications. Click to explore them live.
Home
DU
My Home
Services (1)
employee_Expense_Management_Applic...
3 tables are included in this service. View service detai
Employees
Expenses
Approvals


https://port4004-workspaces-ws-miocc.eu10.applicationstudio.cloud.sap
Update
greytHR
Imported
New folder
Application Development
Project Preview
Expense_Mgmt_C52459
1.0.0
Web Applications (3)
The user will have the following applications. Click to explore them live.
Home
DU
My Home
Approvals
Employees
Expenses
Services (1)
expense_Mgmt_C52459Srv
3 tables are included in this service. View service detai
Employees
Expenses
Approvals

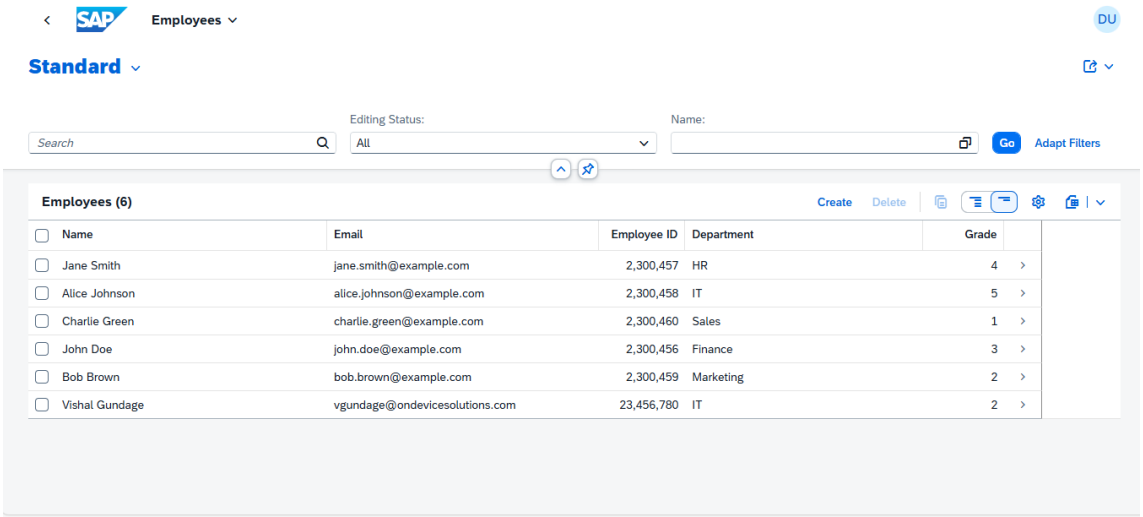

https://port4004-workspaces-ws-miocc.eu10.applicationstudio.cloud.sap
Update
greytHR
Imported
New folder
Web Applications (3)
The user will have the following applications. Click to explore them live.
Home
DU
My Home
Approvals
Manage Approvals
Employees
View Employees
Expenses
Manage Expenses
Services (1)
expense_Mgmt_C52459Srv
3 tables are included in this service. View service detai
Employees
Expenses
Approvals
Search
ENG
IN
17:00
26-12-2025

Test Employees Application

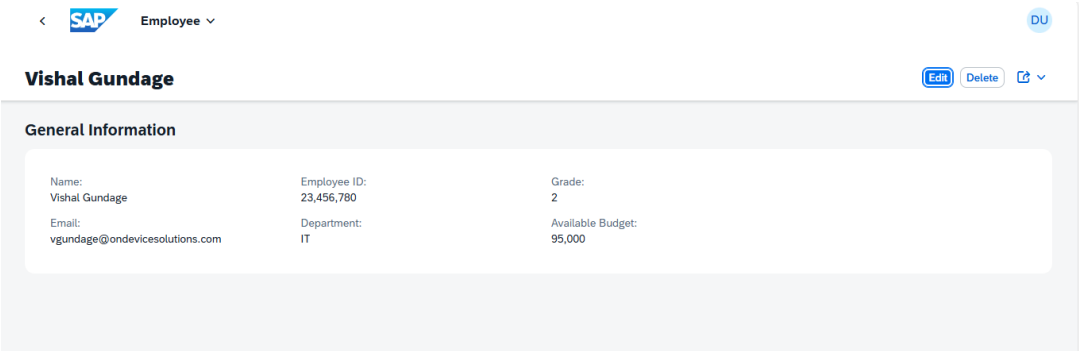
1. Click on the **Employees** tile.



2. The **Employees List Report page** opens – click on the Go button.

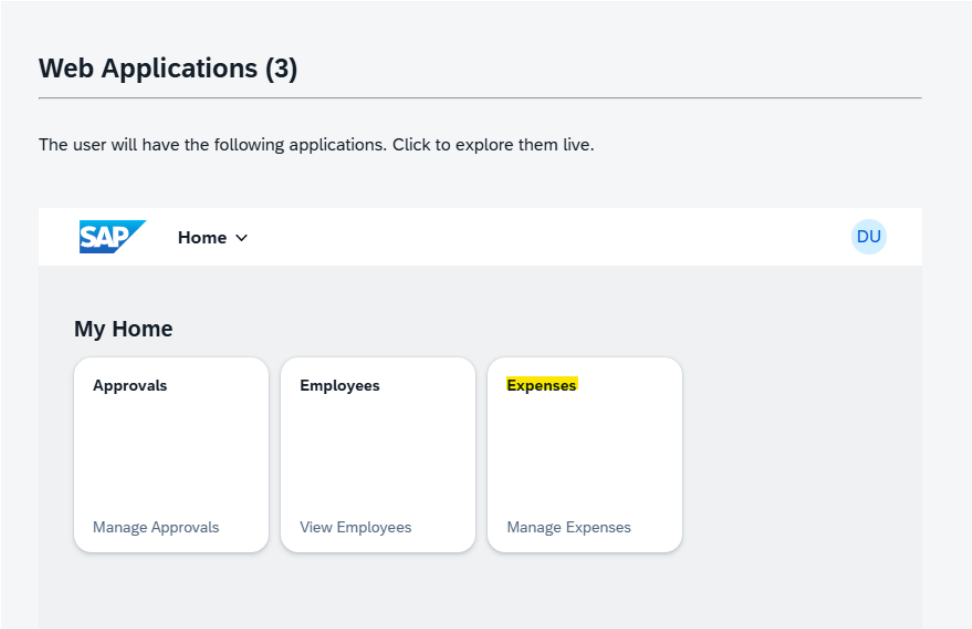


3. Verify employee details

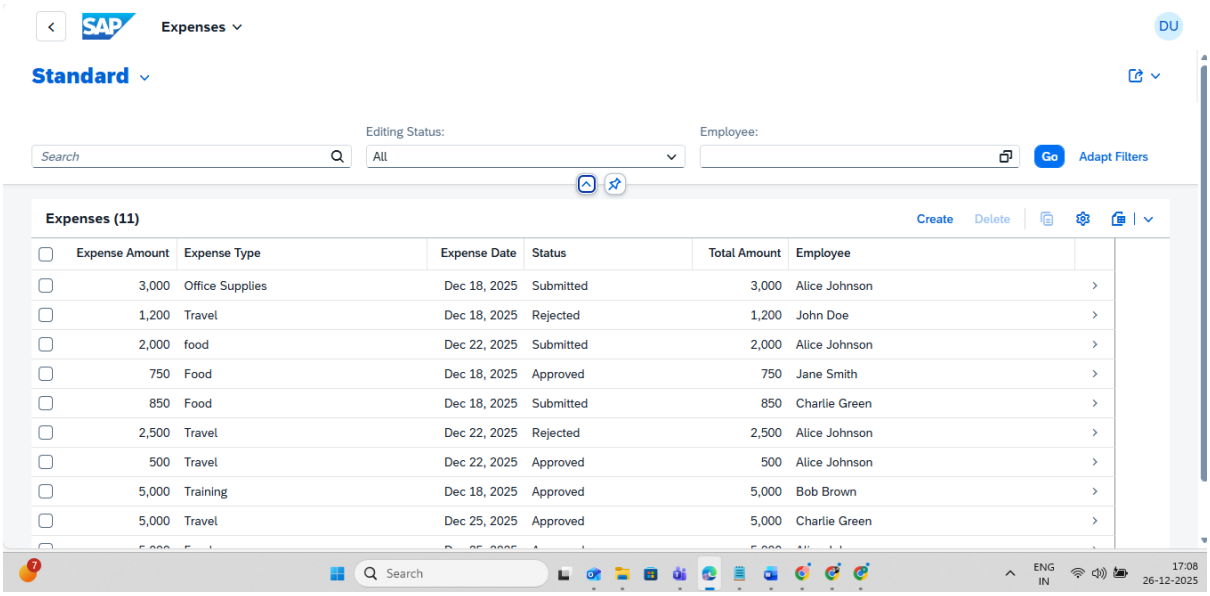


Test Expenses Application

1. Navigate back to the Launchpad.
2. Click on the **Expenses** tile.

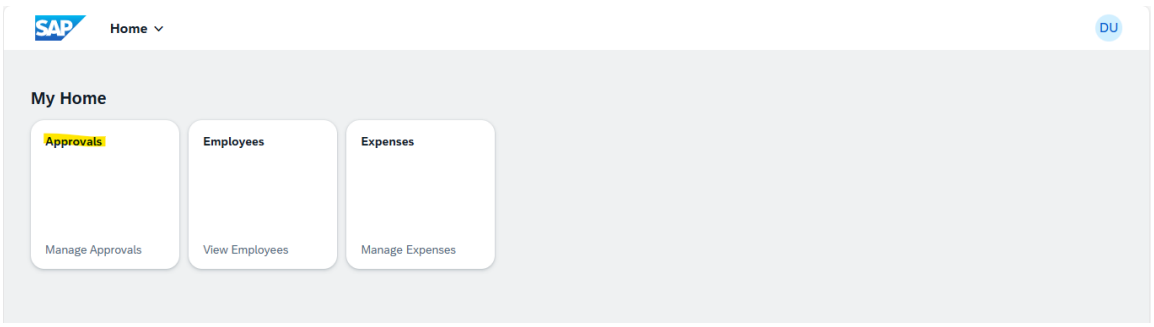


3. The **Expenses List Report page** opens.

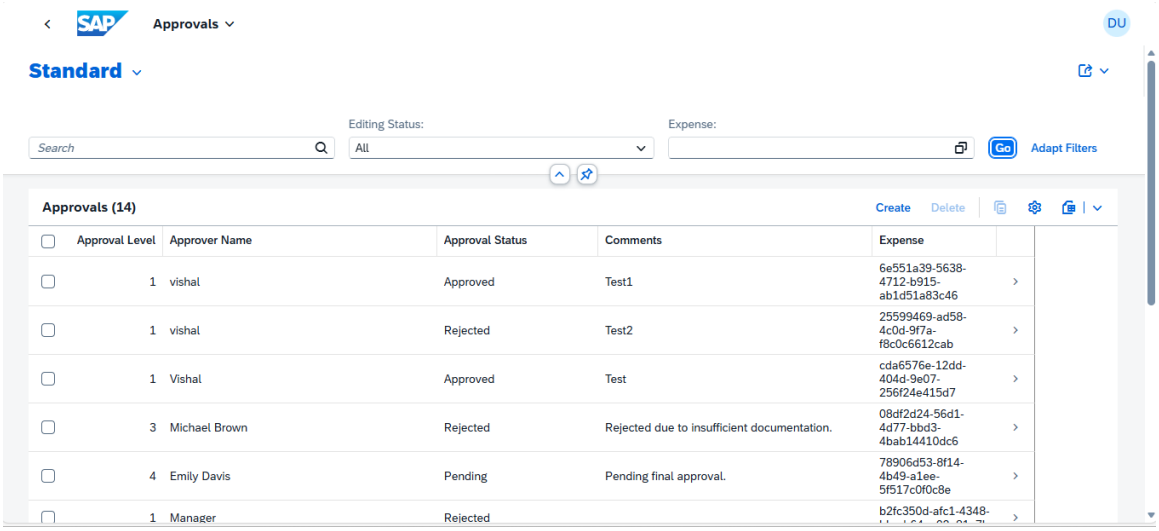


Test Approvals Application

1. Navigate back to the Launchpad.
2. Click on the **Approvals** tile.



3. The **Approvals List Report page** opens.



Conclusion

In this project, we successfully developed an Employee Expense Management Application using SAP Build Code and Joule GenAI. GenAI significantly accelerated application development by generating data models, sample data, backend logic, and UI components with minimal manual effort.

It improved developer productivity and reduced time spent on coding. However, GenAI-generated outputs sometimes required manual refinement to meet exact business requirements and avoid unwanted associations or UI limitations.

Overall, combining GenAI with low-code tools enables faster, smarter application development while still requiring developer validation and control.

Contact us for a personalised no-obligation consultation on how **SAP Build Code with Joule AI** can streamline your workflows, boost productivity, and bring your business ideas to life faster – all while maintaining enterprise-grade standards.

Website: www.ondevicesolutions.com

Email: info@ondevicesolutions.com