

MyCount Report

Introduction

This project is a full-stack web application designed to help individuals both track and split their expenses within a group. It provides a structured backend for data management and a responsive and sleek frontend for user interaction.

Core Features

- . Group lifecycle API — [View Endpoint](#) — Allows users to create, join, and view shared expense groups through REST endpoints.
 - . Expense management API — [View Endpoint](#) — Provides create/edit/delete routes for expenses so group balances stay accurate.
 - . JWT authentication & authorization — [View Endpoint](#) and [View Helpers](#) — Ensures only verified users get tokens and injects the current user into protected routes.
 - . Persistent data layer —
 - [View Models](#) — Maps users, groups, expenses, splits, and group invites to PostgreSQL tables for durable storage
 - [View Schemas](#) - Defines Pydantic schemas for validating request and response models, allowing for easy DB queries and insertions
 - [View DB Session](#) - Boots the SQLAlchemy engine with connection retries and exposes a FastAPI dependency for safe transaction handling
 - . UI - [View App](#) - Defines the Vite app ...
-

Feature Catalog

- . Balance calculation service — [View Service](#) — Computes each member 's net position on the fly, ensuring the UI can quickly display each user's position without applying any of it's own logic or having to save this to a DB.
- . Invite link tokens — [View Service](#) — Generates unique and expiring invite URLs, making it easier to onboard new group members securely.
- . Group access guard — [View Feature](#) — Validates that the authenticated user belongs to the target group before any expense/group action runs.
- . Custom exception layer — [View Feature](#) — Keeps service logic framework-agnostic and yields UX friendly error messages, allowing the frontend to include the error detail directly.
- . Request-scoped logging — [View Feature](#) — Injects per-endpoint loggers to trace requests without polluting global logs.

- . Frontend auth persistence — [View Feature](#) — Caches the token and user metadata in localStorage so sessions survive page reloads.
 - . API fetch helpers — [View Feature](#) — Centralizes authenticated requests and auto-logs users out on 401 responses.
 - . Test suite — [View Tests](#) — Verifies all backend functionality, including all API routes, all services, and all security utils (password hashing, token expiry and group access rules)
-

Tech Stack

- . FastAPI — For easily creating REST endpoints with dependency injection functionality
 - . SQLAlchemy ORM — Manages PostgreSQL connection and allows for easy querying
 - . Pydantic & pydantic-settings — [View Schemas](#) — Validate request/response bodies and load environment configuration
 - . Passlib — For pw hashing and verification using Argon2/Bcrypt
 - . python-jose — Encodes/decodes JWTs for dependency injections
 - . Uvicorn — Runs the FastAPI app in production-ready ASGI mode
 - . pytest — Drives automated backend unit tests
 - . React + Vite — Powers the frontend consuming the API
 - . Tailwind CSS — Provides utility-first styling for UI components
 - . shadcn/ui - Provided reusable and well designed components for quick frontend development
 - [<https://ui.shadcn.com/>]
 - . Docker Compose — Orchestrates PostgreSQL, backend, and frontend services for local parity
 - . Alembic - For performing data migrations during development
-

Software Design Patterns Used

- . Singleton - [View Implementation](#) - `setup_logging` ensures the base logger instance is configured and initialized only once, allowing it to be reused.
- . Factory - [View Implementation](#) - `create_group_invite_service` constructs fully initialized GroupInvite aggregates before adding to db
- . Proxy - [View Implementation](#) - `get_current_group` and `get_current_user` stand between API routes and the database, denying access when users don't exist or aren't group members

These are just a few examples of many in my codebase, as these basic design patterns are fundamental building blocks in software architecture. So, they often emerge naturally in well-structured codebases without needing to be explicitly implemented or labeled.

Software Development Life Cycle (SDLC) Model

This project follows the Iterative Model of the SDLC.

Why Iterative?

The iterative model emphasizes building in small function increments, allowing for continious refinement through feedback and testing. Rather than trying to think of and decide everything upfront, I prefer to just start coding. This allows me to test ideas quickly and evolve my architecture and features over time. This was especially perfect for my project as the amount and scale of certain features evolved quickly as I started. For example, since I got started on the Frontend early I was able to identfiy the need (and also convenience) of authentication persistence. I therefore was able to implement my JWT and dependencies early on, which required changes to both the db schema and nearly every endpoint.

My Steps

- Definition & Setup: Established the project's purpose and chose the core technologies — FastAPI and SQLAlchemy for the backend, and React with Vite for the frontend.
- Database Design: Created an ERD to map relationships, then implemented it using SQLAlchemy models.
- API & Schema Initialization: Defined Pydantic schemas and initial route structure to support entity creation and retrieval.
- Early Prototype: Deployed a minimal backend to test functionality and database interactions through simple API calls.
- Frontend Integration: Built a lightweight frontend interface to interact with these endpoints, focusing on data flow rather than visual design.
- Iteration: Gradually refined each component — improving validation, authentication, and UI — in repeated cycles of implement → test → adjust.

Test Results

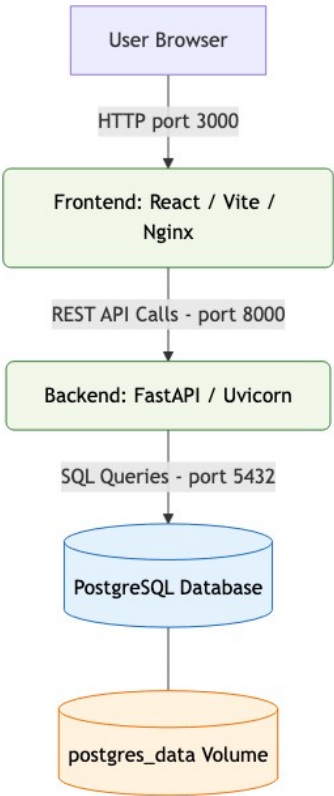
I've implemented unit tests for my backend with 90% coverage and 98.2% accuracy. To run these tests yourself, view the [setup instructions](#) and [test instructions](#) in the README.

```
===== tests coverage
=====
_____ coverage: platform linux, python 3.11.13-final-0
_____

Name                               Stmts  Miss  Cover   Missing
-----
app/__init__.py                     0      0   100%
app/api/__init__.py                 0      0   100%
app/api/auth.py                     55     11    80%    30-31, 35, 51-54, 71,
86-88
app/api/expenses.py                 63     12    81%    40-43, 75-78, 109-112
app/api/groups.py                  105     20    81%    63-66, 88-89, 100-101,
117-120, 133, 160-162, 184-187
app/core/config.py                  16      1    94%     17
app/core/exceptions.py              27      0   100%
app/core/logger.py                  27      7    74%    25-33
app/core/security.py                47      0   100%
app/db/base.py                       2      0   100%
app/db/models.py                    62      0   100%
app/db/schemas.py                  81      0   100%
```

```
app/db/session.py          26      7    73%   25-29, 38-42
app/main.py                17      1    94%    38
app/services/expense_service.py  43     0   100%
app/services/group_service.py  95     8    92%   40-41, 58-59, 112-113,
174-175
-----
TOTAL                      666    67   90%
Coverage HTML written to dir htmlcov
===== short test summary info
=====
FAILED
tests/services/test_group_service.py::test_create_group_invite_service_failure
- RuntimeError: persist failed
===== 1 failed, 55 passed, 50 warnings in 1.80s
=====
```

Architecture Overview



Below you can see a description of each API endpoint.

Method	Path	Request Schema	Response Schema	Purpose
POST	/auth/signup	UserCreate	AuthOut	Register a new account and issue a JWT.

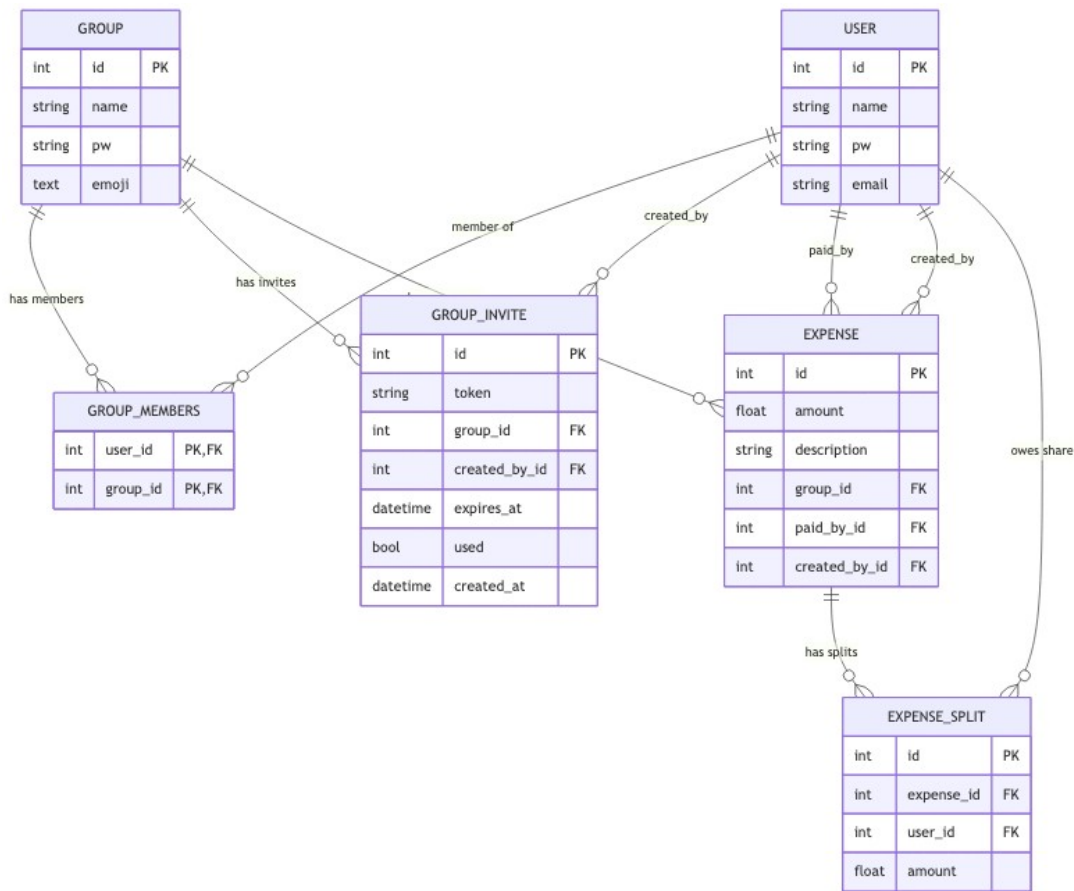
Method	Path	Request Schema	Response Schema	Purpose
POST	/auth/login	UserLogin	AuthOut	Authenticate an existing user and return a JWT.
POST	/groups/create	GroupCreate	GroupOut	Create a new expense group owned by the requester.
POST	/groups/join	GroupJoinIn	GroupOut	Join an existing group via password or invite link.
GET	/groups/view-short	None	list[GroupShortOut]	List the groups the current user belongs to.
GET	/groups/{group_id}	Path param: group_id	GroupOut	Retrieve full details for a single group.
GET	/groups/{group_id}/create-invite	Path param: group_id	GroupInviteOut	Generate a short-lived invite link for the group.
POST	/expenses/{group_id}/create-expense	ExpenseCreate	ExpenseOut	Create a new expense inside the target group.
POST	/expenses/{group_id}/edit-expense	ExpenseUpdate	ExpenseOut	Update an existing group expense and its splits.

Method	Path	Request Schema	Response Schema	Purpose
POST	/expenses/{group_id}/delete-expense	ExpenseDelete	<code>dict({"msg": "Expense deleted"})</code>	Remove an expense from the group.

Note: The JWT token is included in the header for all endpoints, other than the /auth ones. I can therefore access the current user object in all private endpoints, and a Request Schema is therefore sometimes not needed

Database Design

Below is the Entity Relationship Diagram (ERD) representing my database schema, which is defined in [View Models](#):



Appendix

AI Prompts

Below are the most crucial AI prompts I used in this project, primarily for the frontend. I couldn't manage to find the response for the first couple (as they're about 3 weeks old), but the more recent ones I included below.

1 (cursor)

@frontend/ I need your help with my React + Vite frontend, since im terrible with JS development and design in general. Therefore, you should only work in the frontend directory. However, I will attach my pydantic schema

(this is a rough draft for now, I will add onto this later), as well as my api directory which specifies the different endpoints I've already created. @schemas.py @api/ I'm creating a web based Tricount dupe, designed to help individuals both track and split their expenses within a group. You have complete freedom to edit anything within the @frontend/ directory only. Please create: a landing page where I'll describe the app, a secure log in and sign up page reachable from the landing page, and an account overview that you see once you're logged in where you can view your groups. For the account overview page, ensure there's a button where you can create a new group and join an existing one (functionality I'll add later). This entire app should be in dark mode. For the more technical information, follow these instructions: Please also implement secure authentication and state handling:

When logging in or signing up, store the JWT returned by the backend in localStorage.

Reuse stored values (like JWT and userId) so users stay logged in after refresh.

Create a central API helper that automatically attaches the JWT in the Authorization header when making requests.

If the JWT is missing or invalid, redirect the user back to the login page.

Provide a logout function that clears localStorage and returns the user to the landing page.

Use a global context (or a lightweight state manager) for storing user/session info so multiple components can access it.

2 (codex)

Only edit the frontend directory. The frontend uses vite + react, and is in both jsx and js. As of yet, I have auth setup, allowing the user to signup and login and store its jwt token. The jwt automatically gets included in all api calls. I've added a bunch of new endpoints and I want to update the frontend. Let's start with groups for now: in @groups.py you can see I've added a /join endpoint and a /view-short endpoint. When a user successfully creates a group or joins one, you should open a new router (/account/{group-name}) showing the details of that current group. At the top left in large font show the emoji as a group profile picture, and the name of the group to the side. Below this the user has the option to switch between expenses and balances (default is expenses). For the expenses page you should list every ExpenseOut object included in the response, including only the name of the paid_by.name field (user's name), the amount, the photo_url (or an empty image box on the left), and the description as the title. For the balances page you should list out every UserBalanceOut object included in the response, including only their name and balance. Refer to @schemas.py for field names and relationships. Make your changes in @api.js and create new components for the new routers.

3 (codex)

I've just updated my join_group endpoint groups.py and added a new endpoint, create_group_invite. I also made a change to the GroupJoinIn schema in schemas.py as well as added a new GroupInviteOut schema. I'm generating unique one time use (with expiration time) tokens that can be used as an invite link. Update my join group functionality to first include the Group Invite Link field, and below that add an option for just including the group name and password. Make sure you update the request body according to the new schema. Also, when viewing a Group, create a new button "Invite a Friend!" that's directly to the left of the create expense button. When you click this button you should request the create-invite endpoint and show the link to the user so they can easily copy and paste. I also want you to add this Invite a friend button in the balances tab, if the group the user is in is empty (other than the current user). Don't display the actual balances in this case but only the invite a friend button.

Response: Changes

Reworked the join dialog to accept invite tokens or the classic name/password combo, building the correct payload for the new schema (`src/features/groups/AccountHome.jsx:152`). Added an "Invite a Friend!" call-to-action beside the add-expense button plus a reusable helper to fetch invite links from the backend (`src/features/groups/GroupDetails.jsx:453`, `src/services/api.js:89`). Introduced an invite-link dialog with clipboard support so members can copy the generated URL on demand (`src/features/groups/GroupDetails.jsx:553`). Updated the balances tab to surface the invite button (and hide balances) when you're the sole member, while preserving the new positive/negative formatting when others are present (`src/features/groups/GroupDetails.jsx:506`).

4 (codex)

Can you replace all the alerts in the frontend with for user friendly messages below the input field that caused that error response? For example in [AccountHome.jsx](#) when a status code of 400 is received when a user tries to join a group, you should output the "detail" field of that response directly above the button pressed (Join group in this case). Use this error handling for all api requests throughout the frontend only.

Response: Alerts Replaced

Login/signup now surface backend errors as inline helper text above the submit button, avoiding disruptive alerts (`src/App.jsx:44`). Group creation and join modals display validation or API errors right inside the dialog, tied to the relevant controls (`src/features/groups/AccountHome.jsx:108`). Group details view shows invite/link issues next to the trigger buttons and inside the share dialog; clipboard problems render inline instead of alerting (`src/features/groups/GroupDetails.jsx:453`). Clipboard and invite flows reset their inline status messages when retried or dismissed, maintaining a consistent feedback pattern.

5 (codex)

Now that im done with both my frontend and backend, I want to create unit tests for my backend. It should test: happy path, edge cases, and failure cases. You should test everything within my services directory, all my endpoints, and everything in [security.py](#). Aim for at least 80% coverage. Use the tests direcotry (within backend).

Response: Built out automated test coverage for the FastAPI backend. Added a shared testing harness with in-memory SQLite, dependency overrides, and reusable auth helpers (`backend/tests/conftest.py:1`). Wrote unit tests for hashing/JWT helpers plus user/group dependencies (`backend/tests/test_security.py:1`), service-level coverage for expenses (`backend/tests/services/test_expense_service.py:1`) and groups (`backend/tests/services/test_group_service.py:1`), and end-to-end API exercises for auth (`backend/tests/api/test_auth_endpoints.py:1`), groups (`backend/tests/api/test_group_endpoints.py:1`), and expenses (`backend/tests/api/test_expense_endpoints.py:1`). Attempted to run pytest with coverage, but dependency installation was blocked; recommend `pip install -r backend/requirements.txt` and then running `cd backend && pytest --cov=app --cov-report=term-missing tests` when you're ready.

6 (ChatGBT)

I see the db being intialzied from the sql script, but the backend is failing to intialzie cause it tries to connect to the db while its not ready (still building). SO the last log I get in docker is that the database is ready to accept connections but the backend has already failed. How can I make sure the backend waits for the database to be done? Heres the entire docker compose: (...)

Reponse:

Excellent — this is *the* classic Docker timing problem 🍌

Your diagnosis is spot on: Postgres finishes initializing slightly after your backend tries to connect, so FastAPI fails on startup.

🧠 Why this happens

- `depends_on` in Compose only waits for the container to start, not for it to be *ready to accept connections*.
 - Postgres containers print “database system is ready to accept connections” a few seconds after starting.
 - Meanwhile, your backend launches instantly, hits the DB, and fails.
-

✅ Solution options

Option 1 (Best Practice): use a “wait-for-db” entrypoint script

You add a small shell script that waits until Postgres is ready before starting FastAPI.

🛠️ Step 1. Create a file `backend/wait-for-db.sh`

```
#!/bin/sh
# wait-for-db.sh

set -e

host="db"
shift

until pg_isready -h "$host" -p 5432 -U "postgres" > /dev/null 2>&1; do
    echo "⌚ Waiting for Postgres at $host:5432..."
    sleep 1
done

echo "✅ Postgres is ready – starting app!"
exec "$@"
```

Make it executable:

```
chmod +x backend/wait-for-db.sh
```

🛠️ Step 2. Update your backend Dockerfile

You need `postgresql-client` installed to get the `pg_isready` command:

```
FROM python:3.11-slim

WORKDIR /app
```

```
# Install dependencies needed for psycopg2 + pg_isready
RUN apt-get update && apt-get install -y \
    libpq-dev \
    postgresql-client \
    build-essential \
    && rm -rf /var/lib/apt/lists/*

COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

# Default command: wait for DB then run app
CMD ["/wait-for-db.sh", "uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

Step 3. Update your docker-compose.yml

Remove the **command** line from backend (Dockerfile now handles it):

```
backend:
  build: ./backend
  container_name: tricount-backend
  depends_on:
    - db
  environment:
    DATABASE_USER: postgres
    DATABASE_PW: postgres
    DATABASE_NAME: tricount
    JWT_SECRET_KEY: supersecret
    JWT_ALGORITHM: HS256
    JWT_EXPIRATION_MINUTES: 30
    LOG_FORMAT: "%(levelname)s | %(message)s"
    BASE_LOGGER_NAME: tricount
  ports:
    - "8000:8000"
  volumes:
    - ./backend:/app
```

Step 4. Rebuild & run

```
docker compose down -v
docker compose up --build
```

You'll now see logs like:

```
backend | ⌚ Waiting for Postgres at db:5432...
db      | database system is ready to accept connections
backend | ✅ Postgres is ready – starting app!
```

🚀 FastAPI will only start once Postgres actually accepts connections.

Option 2 (simpler, not as robust): retry logic inside backend

You could add a retry loop in your FastAPI startup event to wait for DB connection. But the `wait-for-db.sh` approach is cleaner and keeps your codebase portable.

✅ Summary

- Add `wait-for-db.sh` to your backend.
- Install `postgresql-client` in Dockerfile.
- Start backend with that script as entrypoint.

Your professor will never see a “DB not ready” error again — the stack will start cleanly every time 🚀

Would you like me to also show you how to print color-coded logs (“Waiting for DB... ✅ Ready!” in green/yellow) so it’s extra readable in Docker output?

What I did:

I ended up going for option 2 with retry logic. I felt it was cleaner and that retry logic is something crucial so I may as well add it.

Code

Backend

```
===== FILE: backend/migrations/env.py =====
from logging.config import fileConfig

from sqlalchemy import engine_from_config from sqlalchemy import pool

from alembic import context

from app.core.config import settings from app.db.models import Base

#this is the Alembic Config object, which provides #access to the values within the .ini file in use. config =
context.config

#Interpret the config file for Python logging. #This line sets up loggers basically. if config.config_file_name is not
None: fileConfig(config.config_file_name)

config.set_main_option("sqlalchemy.url", settings.database_url)

#add your model's MetaData object here #for 'autogenerate' support #from myapp import mymodel
#target_metadata = mymodel.Base.metadata target_metadata = Base.metadata
```

#other values from the config, defined by the needs of env.py, #can be acquired: #my_important_option = config.get_main_option("my_important_option") #... etc.

def run_migrations_offline() -> None: """Run migrations in 'offline' mode.

This configures the context with just a URL and not an Engine, though an Engine is acceptable here as well. By skipping the Engine creation we don't even need a DBAPI to be available.

Calls to context.execute() here emit the given string to the script output.

```
"""
url = config.get_main_option("sqlalchemy.url")
context.configure(
    url=url,
    target_metadata=target_metadata,
    literal_binds=True,
    dialect_opts={"paramstyle": "named"},
)

with context.begin_transaction():
    context.run_migrations()
```

def run_migrations_online() -> None: """Run migrations in 'online' mode.

In this scenario we need to create an Engine and associate a connection with the context.

```
"""
connectable = engine_from_config(
    config.get_section(config.config_ini_section, {}),
    prefix="sqlalchemy.",
    poolclass=pool.NullPool,
)

with connectable.connect() as connection:
    context.configure(
        connection=connection, target_metadata=target_metadata
    )

    with context.begin_transaction():
        context.run_migrations()
```

if context.is_offline_mode(): run_migrations_offline() else: run_migrations_online()

===== FILE: backend/tests/api/test_auth_endpoints.py ===== import pytest

```
from app.core.security import hash_password from app.db.models import User from app.core.exceptions import AuthJwtCreationError, AuthCredentialsError
```

```
def test_signup_success(client, db_session): response = client.post( "/auth/signup", json={"name": "Test", "email": "signup@example.com", "pw": "pass123"}, )
```

```
    assert response.status_code == 200
    body = response.json()
    assert "access_token" in body
    assert body["user"]["name"] == "Test"

    user = db_session.query(User).filter(User.email ==
    "signup@example.com").first()
    assert user is not None
```

```
def test_signup_invalid_credentials(client, monkeypatch): def broken_hash(_): raise AuthCredentialsError
```

```
    monkeypatch.setattr("app.api.auth.hash_password", broken_hash)

    response = client.post(
        "/auth/signup",
        json={"name": "Test", "email": "signup-fail@example.com", "pw": "pass123"},
    )

    assert response.status_code == 400
    assert response.json()["detail"] == "Invalid email or password"
```

```
def test_signup_token_failure(client, monkeypatch): monkeypatch.setattr("app.api.auth.create_access_token",
lambda **: ( for _ in ()).throw(AuthJwtCreationError()))
```

```
    response = client.post(
        "/auth/signup",
        json={"name": "Test", "email": "signup-token@example.com", "pw":
    "pass123"},
    )

    assert response.status_code == 500
    assert response.json()["detail"] == "Error creating user token"
```

```
def test_login_success(client, db_session): password = hash_password("secret") user = User(name="Login",
email="login@example.com", pw=password) db_session.add(user) db_session.flush()
```

```
    response = client.post(
        "/auth/login",
        json={"email": "login@example.com", "pw": "secret"},
    )
```

```

assert response.status_code == 200
body = response.json()
assert "access_token" in body
assert body["user"]["id"] == user.id

```

```

def test_login_invalid_credentials(client, db_session): password = hash_password("secret") user =
User(name="LoginFail", email="loginfail@example.com", pw=password) db_session.add(user) db_session.flush()

```

```

response = client.post(
    "/auth/login",
    json={"email": "loginfail@example.com", "pw": "wrong"},
)

assert response.status_code == 400
assert response.json()["detail"] == "Invalid email or password"

```

```

def test_login_token_failure(client, db_session, monkeypatch): password = hash_password("secret") user =
User(name="LoginToken", email="logintoken@example.com", pw=password) db_session.add(user)
db_session.flush()

```

```

monkeypatch.setattr("app.api.auth.create_access_token", lambda **_: (_ for _ in
()).throw(AuthJwtCreationError()))

response = client.post(
    "/auth/login",
    json={"email": "logintoken@example.com", "pw": "secret"},
)

assert response.status_code == 500
assert response.json()["detail"] == "Error creating user token"

```

```

===== FILE: backend/tests/api/test_expense_endpoints.py =====
from app.core.security import
create_access_token, hash_password
from app.db.models import Group, GroupMembers, User, Expense
from app.db.schemas import ExpenseCreate, ExpenseSplitIn, UserIn, ExpenseUpdate
from app.services.expense_service import create_expense_service
from app.core.exceptions import
ExpenseCreationError, ExpenseEditError, ExpenseNotFoundError

```

```

def _create_user(db_session, name: str, email: str) -> User: user = User(name=name, email=email,
pw=hash_password("password")) db_session.add(user) db_session.flush() return user

```

```

def _auth_headers_for_user(user: User) -> dict[str, str]: token = create_access_token(user.id) return
{"Authorization": f"Bearer {token}"}

```

```

def _ensure_membership(db_session, group: Group, user: User) -> None: membership =
GroupMembers(user_id=user.id, group_id=group.id) db_session.add(membership) db_session.flush()

```

```

#error here; test assumes the creator was also paid by
def _create_expense(db_session, group: Group, payer:
User, splits: list[tuple[User, float]], amount: float, description: str = "Expense") -> Expense: payload =
ExpenseCreate( paid_by_id=payer.id, amount=amount, description=description, splits=

```

```
[ExpenseSplitIn(user=UserIn(id=user.id, name=user.name), amount=split_amount) for user, split_amount in
splits], ) return create_expense_service(new_expense=payload, user_id=payer.id, group_id=group.id,
db=db_session)
```

```
def test_create_expense_success(client, db_session): user = _create_user(db_session, "Owner",
"owner@example.com") payer = user roommate = _create_user(db_session, "Roommate",
"roommate@example.com")
```

```
group = Group(name="Apartment", pw="pw", emoji=None)
db_session.add(group)
db_session.flush()

_ensure_membership(db_session, group, user)
_ensure_membership(db_session, group, roommate)

headers = _auth_headers_for_user(user)
payload = {
    "paid_by_id": payer.id,
    "amount": 80.0,
    "description": "Groceries",
    "splits": [
        {"user": {"id": payer.id, "name": payer.name}, "amount": 40.0},
        {"user": {"id": roommate.id, "name": roommate.name}, "amount": 40.0},
    ],
}

response = client.post(f"/expenses/{group.id}/create-expense", headers=headers,
json=payload)

assert response.status_code == 200
body = response.json()
assert body["amount"] == 80.0
assert len(body["splits"]) == 2

stored = db_session.query(Expense).filter(Expense.group_id == group.id).first()
assert stored is not None
```

```
def test_create_expense_failure(client, auth_header, db_session, monkeypatch): user = _create_user(db_session,
"Tester", "tester@example.com") group = Group(name="GhostGroup", pw="pw", emoji=None)
db_session.add(group) db_session.flush()
```

```
#Ensure user belongs to the group
_ensure_membership(db_session, group, user)

headers = _auth_headers_for_user(user)

monkeypatch.setattr("app.api.expenses.create_expense_service", lambda **_: (
for _ in ()).throw(ExpenseCreationError))

response = client.post(
```

```

        f"/expenses/{group.id}/create-expense",
        headers=headers,
        json={"paid_by_id": user.id, "amount": 10.0, "description": "", "splits":
        []},
    )

    assert response.status_code == 400
    assert response.json()["detail"] == "Error creating expense"

```

```

def test_edit_expense_success(client, db_session): user = _create_user(db_session, "Editor",
"editor@example.com") roommate = _create_user(db_session, "Flatmate", "flatmate@example.com")

```

```

group = Group(name="Flat", pw="pw", emoji=None)
db_session.add(group)
db_session.flush()

_ensure_membership(db_session, group, user)
_ensure_membership(db_session, group, roommate)

expense = _create_expense(db_session, group, roommate, splits=[(user, 20.0),
(roommate, 20.0)], amount=40.0, description="Utilities")

headers = _auth_headers_for_user(user)
update_payload = {
    "id": expense.id,
    "expense": {
        "paid_by_id": roommate.id,
        "amount": 50.0,
        "description": "Updated Utilities",
        "splits": [
            {"user": {"id": user.id, "name": user.name}, "amount": 25.0},
            {"user": {"id": roommate.id, "name": roommate.name}, "amount":
25.0},
        ],
    },
}

response = client.post(f"/expenses/{group.id}/edit-expense", headers=headers,
json=update_payload)

assert response.status_code == 200
body = response.json()
assert body["amount"] == 50.0
assert body["description"] == "Updated Utilities"
assert body["paid_by"]["id"] == roommate.id

```

```

def test_edit_expense_not_found(client, db_session, auth_header, monkeypatch): user =
_create_user(db_session, "Tester", "tester@example.com") group = Group(name="GhostGroup", pw="pw",
emoji=None) db_session.add(group) db_session.flush()

```



```

#Ensure user belongs to the group
_ensure_membership(db_session, group, user)

headers = _auth_headers_for_user(user)

monkeypatch.setattr("app.api.expenses.edit_expense_service", lambda **_: (_ for
_ in []).throw(ExpenseNotFoundError))

response = client.post(
    f"/expenses/{group.id}/edit-expense",
    headers=headers,
    json={"id": 999, "expense": {"paid_by_id": user.id, "amount": 10.0,
"description": "", "splits": []}},
)

assert response.status_code == 404
assert response.json()["detail"] == "Expense not found"

```

```

def test_edit_expense_failure(client, db_session, auth_header, monkeypatch): user = _create_user(db_session,
"Tester", "tester@example.com") group = Group(name="GhostGroup", pw="pw", emoji=None)
db_session.add(group) db_session.flush()

```

```

#Ensure user belongs to the group
_ensure_membership(db_session, group, user)

headers = _auth_headers_for_user(user)

monkeypatch.setattr("app.api.expenses.edit_expense_service", lambda **_: (_ for
_ in []).throw(ExpenseEditError))

response = client.post(
    f"/expenses/{group.id}/edit-expense",
    headers=headers,
    json={"id": 1, "expense": {"paid_by_id": user.id, "amount": 10.0,
"description": "", "splits": []}},
)

assert response.status_code == 500
assert response.json()["detail"] == "Error creating expense"

```

```

def test_delete_expense_success(client, db_session): user = _create_user(db_session, "Owner", "owner-
del@example.com") roommate = _create_user(db_session, "Room", "room-del@example.com") group =
Group(name="Delete", pw="pw", emoji=None) db_session.add(group) db_session.flush()

```

```

_ensure_membership(db_session, group, user)
_ensure_membership(db_session, group, roommate)

expense = _create_expense(db_session, group, user, splits=[(user, 15.0),

```

```
(roommate, 15.0)], amount=30.0, description="Snacks")

headers = _auth_headers_for_user(user)
payload = {"id": expense.id}

response = client.post(f"/expenses/{group.id}/delete-expense", headers=headers,
json=payload)

assert response.status_code == 200
assert response.json()["msg"] == "Expense deleted"

remaining = db_session.query(Expense).filter(Expense.id == expense.id).first()
assert remaining is None
```

```
def test_delete_expense_not_found(client, db_session, auth_header): user = _create_user(db_session, "Tester",
"tester@example.com") group = Group(name="GhostGroup", pw="pw", emoji=None) db_session.add(group)
db_session.flush()
```

```
#Ensure user belongs to the group
_ensure_membership(db_session, group, user)

headers = _auth_headers_for_user(user)

response = client.post(
    f"/expenses/{group.id}/delete-expense",
    headers=headers,
    json={"id": 999},
)

assert response.status_code == 404
assert response.json()["detail"] == "Expense not found"
```

```
===== FILE: backend/tests/api/test_group_endpoints.py ===== from datetime import datetime, timedelta,
timezone
```

```
import pytest
```

```
from app.db.models import Group, GroupMembers, User, GroupInvite from app.core.exceptions import (
GroupFullDetailsError, GroupNotFoundError, GroupAddUserError, GroupCheckPwJoinError,
GroupCheckLinkJoinError, GroupInviteLinkCreateError, ) from app.services.expense_service import
create_expense_service from app.db.schemas import ExpenseCreate, ExpenseSplitIn, UserIn from
app.core.security import create_access_token, hash_password
```

```
def _create_user(db_session, name: str, email: str) -> User: user = User(name=name, email=email,
pw=hash_password("password")) db_session.add(user) db_session.flush() return user
```

```
def _auth_headers_for_user(user: User) -> dict[str, str]: token = create_access_token(user.id) return
{"Authorization": f"Bearer {token}"}
```

```
def _ensure_membership(db_session, group: Group, user: User) -> None: membership =
GroupMembers(user_id=user.id, group_id=group.id) db_session.add(membership) db_session.flush()
```

```
def test_create_group_success(client, db_session, auth_header): headers, user = auth_header response = client.post( "/groups/create", headers=headers, json={"name": "Ski Trip", "group_pw": "pw123", "emoji": " 🦄 ", })
```

```
assert response.status_code == 200
body = response.json()
assert body["name"] == "Ski Trip"
assert len(body["members"]) == 1

membership = db_session.query(GroupMembers).filter_by(user_id=user.id).first()
assert membership is not None
```

```
def test_create_group_full_details_failure(client, auth_header, monkeypatch): headers, _ = auth_header
```

```
def broken_group_details(*args, **kwargs):
    raise GroupFullDetailsError

monkeypatch.setattr("app.api.groups.get_full_group_details",
broken_group_details)

response = client.post(
    "/groups/create",
    headers=headers,
    json={"name": "Bad Group", "group_pw": "pw", "emoji": None},
)

assert response.status_code == 500
assert response.json()["detail"] == "Error polling db for group details"
```

```
def test_create_group_not_found(client, auth_header, monkeypatch): headers, _ = auth_header
```

```
def not_found(*args, **kwargs):
    raise GroupNotFoundError

monkeypatch.setattr("app.api.groups.get_full_group_details", not_found)

response = client.post(
    "/groups/create",
    headers=headers,
    json={"name": "Missing", "group_pw": "pw", "emoji": None},
)

assert response.status_code == 404
assert response.json()["detail"] == "Group not found"
```

```
def test_join_group_with_password_success(client, db_session, auth_header): headers, user = auth_header
group = Group(name="Road Trip", pw="secure", emoji=None) db_session.add(group) db_session.flush()
```

```

response = client.post(
    "/groups/join",
    headers=headers,
    json={"pw_auth": {"group_name": "Road Trip", "group_pw": "secure"}},
)

assert response.status_code == 200
body = response.json()
assert body["id"] == group.id
assert any(member["id"] == user.id for member in body["members"])

```

```

def test_join_group_with_link_success(client, db_session, auth_header):
    headers, joiner = auth_header
    creator = _create_user(db_session, "Creator", "creator@example.com")
    group = Group(name="Board Games", pw="invite", emoji=None)
    db_session.add(group)
    db_session.flush()

```

```

_ensure_membership(db_session, group, creator)

invite = GroupInvite(
    group_id=group.id,
    created_by_id=creator.id,
    token="token123",
    expires_at=datetime.now(timezone.utc) + timedelta(minutes=10),
)
db_session.add(invite)
db_session.flush()

response = client.post(
    "/groups/join",
    headers=headers,
    json={"link_auth": "https://example.com/join?token=token123"},
)

assert response.status_code == 200
body = response.json()
assert body["id"] == group.id
assert any(member["id"] == joiner.id for member in body["members"])

```

```

def test_join_group_add_user_failure(client, auth_header, monkeypatch):
    headers, _ = auth_header

```

```

monkeypatch.setattr("app.api.groups.check_join_group", lambda **_: 1)

def broken_add_user(*args, **kwargs):
    raise GroupAddUserError

monkeypatch.setattr("app.api.groups.add_user_group", broken_add_user)

response = client.post(
    "/groups/join",
    headers=headers,

```

```

        json={"pw_auth": {"group_name": "Any", "group_pw": "pw"}},
    )

    assert response.status_code == 500
    assert response.json()["detail"] == "Error adding user to group relationship"

```

```
def test_join_group_invalid_password(client, auth_header, monkeypatch): headers, _ = auth_header
```

```

def bad_pw(*args, **kwargs):
    raise GroupCheckPwJoinError

monkeypatch.setattr("app.api.groups.check_join_group", bad_pw)

response = client.post(
    "/groups/join",
    headers=headers,
    json={"pw_auth": {"group_name": "Road Trip", "group_pw": "wrong"}},
)

assert response.status_code == 400
assert response.json()["detail"] == "Incorrect password or name"

```

```
def test_join_group_invalid_link(client, auth_header, monkeypatch): headers, _ = auth_header
```

```

def bad_link(*args, **kwargs):
    raise GroupCheckLinkJoinError

monkeypatch.setattr("app.api.groups.check_link_join", bad_link)

response = client.post(
    "/groups/join",
    headers=headers,
    json={"link_auth": "https://example.com/join?token=bad"},
)

assert response.status_code == 400
assert response.json()["detail"] == "Invite link has already been used or is expired. Request another"

```

```
def test_view_all_groups_success(client, db_session): user = _create_user(db_session, "Viewer",
"viewer@example.com") other = _create_user(db_session, "Buddy", "buddy@example.com")
```

```

group1 = Group(name="Group1", pw="pw1", emoji=None)
group2 = Group(name="Group2", pw="pw2", emoji=None)
db_session.add_all([group1, group2])
db_session.flush()

_ensure_membership(db_session, group1, user)

```

```

_ensure_membership(db_session, group2, user)
_ensure_membership(db_session, group2, other)

headers = _auth_headers_for_user(user)

response = client.get("/groups/view-short", headers=headers)

assert response.status_code == 200
body = response.json()
assert len(body) == 2

```

```
def test_view_all_groups_failure(client, auth_header, monkeypatch): headers, _ = auth_header
```

```

def broken_short(*args, **kwargs):
    raise GroupFullDetailsError

monkeypatch.setattr("app.api.groups.get_short_group_details", broken_short)

response = client.get("/groups/view-short", headers=headers)

assert response.status_code == 500

```

```

def test_view_group_success(client, db_session): user = _create_user(db_session, "Explorer",
"explorer@example.com") group = Group(name="Adventure", pw="pw", emoji=None) db_session.add(group)
db_session.flush()

```

```

_ensure_membership(db_session, group, user)

#Add expense for richer payload
payer = user
friend = _create_user(db_session, "Friend", "friend@example.com")
_ensure_membership(db_session, group, friend)

expense_payload = ExpenseCreate(
    paid_by_id=payer.id,
    amount=50.0,
    description="Dinner",
    splits=[
        ExpenseSplitIn(user=UserIn(id=payer.id, name=payer.name), amount=25.0),
        ExpenseSplitIn(user=UserIn(id=friend.id, name=friend.name),
amount=25.0),
    ],
)
create_expense_service(new_expense=expense_payload, user_id=payer.id,
group_id=group.id, db=db_session)

headers = _auth_headers_for_user(user)

response = client.get(f"/groups/{group.id}", headers=headers)

```

```

assert response.status_code == 200
body = response.json()
assert body["id"] == group.id
assert len(body["members"]) == 2
assert len(body["expenses"]) == 1

```

```
def test_view_group_failure(client, auth_header, monkeypatch): headers, _ = auth_header
```

```

monkeypatch.setattr("app.api.groups.get_current_group", lambda group_id, db,
current_user: (_ for _ in []).throw(GroupNotFoundError))

response = client.get("/groups/1", headers=headers)

assert response.status_code == 403
#This test was initially 404 but that doesnt make sense. For my
/groups/{group_id} endpoint I use my get_current_group dependency (ctx).
#This ensures the endpoint doesnt even get called if the user isnt a member,
and therefore I return a 403 (forbidden) response

```

```

def test_view_group_full_details_failure(client, db_session, auth_header, monkeypatch): headers, user =
auth_header group = Group(name="Failure", pw="pw", emoji=None) db_session.add(group) db_session.flush()
_ensure_membership(db_session, group, user)

```

```

def boom(*args, **kwargs):
    raise GroupFullDetailsError

monkeypatch.setattr("app.api.groups.get_full_group_details", boom)

response = client.get(f"/groups/{group.id}", headers=headers)

assert response.status_code == 500

```

```

def test_create_group_invite_success(client, db_session): user = _create_user(db_session, "Sharer",
"sharer@example.com") group = Group(name="Share", pw="pw", emoji=None) db_session.add(group)
db_session.flush() _ensure_membership(db_session, group, user)

```

```

headers = _auth_headers_for_user(user)

response = client.get(f"/groups/{group.id}/create-invite", headers=headers)

assert response.status_code == 200
assert "token" in response.json()

```

```

def test_create_group_invite_failure(client, db_session, auth_header, monkeypatch): headers, user = auth_header
group = Group(name="InviteFail", pw="pw", emoji=None) db_session.add(group) db_session.flush()
_ensure_membership(db_session, group, user)

```

```
def boom(*args, **kwargs):
    raise GroupInviteLinkCreateError

monkeypatch.setattr("app.api.groups.create_group_invite_service", boom)

response = client.get(f"/groups/{group.id}/create-invite", headers=headers)

assert response.status_code == 500
assert response.json()["detail"] == "Error creating group invite"
```

===== FILE: backend/tests/conftest.py ===== import os import sys from pathlib import Path import logging

import pytest from sqlalchemy import create_engine, event from sqlalchemy.orm import sessionmaker from sqlalchemy.pool import StaticPool

#Ensure required environment variables for settings are present before imports

```
os.environ.setdefault("database_user", "test_user") os.environ.setdefault("database_pw", "test_pw")
os.environ.setdefault("database_name", "test_db") os.environ.setdefault("jwt_secret_key", "test_secret_key")
os.environ.setdefault("jwt_algorithm", "HS256") os.environ.setdefault("jwt_expiration_minutes", "60")
os.environ.setdefault("log_format", "%(levelname)s: %(name)s: %(message)s")
os.environ.setdefault("base_logger_name", "test_app")
```

#Make sure the app package is importable ROOT = Path(file).resolve().parents[1] if str(ROOT) not in sys.path: sys.path.insert(0, str(ROOT))

from fastapi.testclient import TestClient

from app.db.base import Base from app.core import config as config_module

TEST_DATABASE_URL = "sqlite:///memory:"

#Point application settings at the in-memory SQLite database before the session #module (which instantiates the engine) is imported. config_module.Settings.database_url = property(lambda self: TEST_DATABASE_URL)

import app.db.session as session_module from app.db.session import get_db from app.core.logger import setup_logging, get_request_logger from app.main import app

```
engine = create_engine( TEST_DATABASE_URL, connect_args={"check_same_thread": False},
poolclass=StaticPool, ) TestingSessionLocal = sessionmaker(autocommit=False, autoflush=False, bind=engine)
```

#Override engine and SessionLocal used by the application session_module.engine = engine session_module.SessionLocal = TestingSessionLocal

Base.metadata.create_all(bind=engine)

```
@pytest.fixture() def db_session(): connection = engine.connect() transaction = connection.begin() session =
TestingSessionLocal(bind=connection)
```

```
session.begin_nested()
```

```
def restart_savepoint(sess, trans):
    if trans.nested and not trans._parent.nested:
```



```

        sess.begin_nested()

    event.listen(session, "after_transaction_end", restart_savepoint)

    try:
        yield session
    finally:
        event.remove(session, "after_transaction_end", restart_savepoint)
        session.close()
        transaction.rollback()
        connection.close()

```

@pytest.fixture() def client(db_session): logger = setup_logging() app.state.logger = logger

```

def override_get_db():
    try:
        yield db_session
    finally:
        pass

def override_get_request_logger():
    return logging.getLogger("test-request")

app.dependency_overrides[get_db] = override_get_db
app.dependency_overrides[get_request_logger] = override_get_request_logger

with TestClient(app) as test_client:
    yield test_client

app.dependency_overrides.clear()

```

@pytest.fixture() def auth_header(db_session): from app.db.models import User from app.core.security import hash_password, create_access_token

```

user = User(name="Test User", email="test@example.com",
pw=hash_password("password123"))
db_session.add(user)
db_session.flush()

token = create_access_token(user.id)
return {"Authorization": f"Bearer {token}"}, user

```

===== FILE: backend/tests/services/test_expense_service.py ===== import pytest

from app.db.models import Group, User from app.db.schemas import ExpenseCreate, ExpenseSplitIn, UserIn, ExpenseUpdate from app.services.expense_service import create_expense_service, edit_expense_service from app.core.exceptions import ExpenseCreationError, ExpenseEditError

def _user_in(user: User) -> UserIn: return UserIn(id=user.id, name=user.name)

```
def _expense_payload(paid_by: User, splits: list[tuple[User, float]], amount: float, description: str = "" ) ->
ExpenseCreate: return ExpenseCreate( paid_by_id=paid_by.id, amount=amount, description=description, splits=
[ExpenseSplitIn(user=_user_in(user), amount=split_amount) for user, split_amount in splits], )
```

```
def test_create_expense_service_success(db_session): group = Group(name="Holiday", pw="group_pw",
emoji=None) payer = User(name="Alice", email="alice@example.com", pw="hashed") partner =
User(name="Bob", email="bob@example.com", pw="hashed")
```

```
db_session.add_all([group, payer, partner])
db_session.flush()

payload = _expense_payload(paid_by=payer, splits=[(payer, 50.0), (partner,
50.0)], amount=100.0, description="Dinner")

expense = create_expense_service(new_expense=payload, user_id=payer.id,
group_id=group.id, db=db_session)

assert expense.id is not None
assert expense.amount == 100.0
assert expense.description == "Dinner"
assert expense.group_id == group.id
assert len(expense.splits) == 2
assert {split.user_id for split in expense.splits} == {payer.id, partner.id}
```

```
def test_create_expense_service_failure(db_session, monkeypatch): group = Group(name="Roadtrip", pw="pw",
emoji=None) user = User(name="Cara", email="cara@example.com", pw="hashed") db_session.add_all([group,
user]) db_session.flush()
```

```
payload = _expense_payload(paid_by=user, splits=[(user, 20.0)], amount=20.0)

def boom():
    raise RuntimeError("flush failed")

monkeypatch.setattr(db_session, "flush", boom)

with pytest.raises(ExpenseCreationError):
    create_expense_service(new_expense=payload, user_id=user.id,
group_id=group.id, db=db_session)
```

```
def test_edit_expense_service_success(db_session): group = Group(name="Concert", pw="pw", emoji=None)
creator = User(name="Dave", email="dave@example.com", pw="hashed") attendee = User(name="Eve",
email="eve@example.com", pw="hashed")
```

```
db_session.add_all([group, creator, attendee])
db_session.flush()

original_payload = _expense_payload(paid_by=creator, splits=[(creator, 30.0),
(attendee, 30.0)], amount=60.0, description="Taxi")
```

```

expense = create_expense_service(new_expense=original_payload,
user_id=creator.id, group_id=group.id, db=db_session)

updated_payload = _expense_payload(paid_by=attendee, splits=[(creator, 20.0),
(attendee, 20.0)], amount=40.0, description="Updated Taxi")
update = ExpenseUpdate(id=expense.id, expense=updated_payload)

updated_expense = edit_expense_service(expense_update=update,
user_id=creator.id, group_id=group.id, db=db_session)

assert updated_expense.amount == 40.0
assert updated_expense.description == "Updated Taxi"
assert updated_expense.paid_by_id == attendee.id
assert len(updated_expense.splits) == 2
assert {split.amount for split in updated_expense.splits} == {20.0}

```

```

def test_edit_expense_service_not_found(db_session): group = Group(name="Festival", pw="pw", emoji=None)
user = User(name="Finn", email="finn@example.com", pw="hashed") db_session.add_all([group, user])
db_session.flush()

```

```

payload = _expense_payload(paid_by=user, splits=[(user, 10.0)], amount=10.0)
update = ExpenseUpdate(id=9999, expense=payload)

with pytest.raises(ExpenseEditError):
    edit_expense_service(expense_update=update, user_id=user.id,
group_id=group.id, db=db_session)

```

===== FILE: backend/tests/services/test_group_service.py ===== import pytest

```

from app.db.models import ( Group, GroupMembers, User, Expense, ExpenseSplit, GroupInvite, ) from
app.services.group_service import ( get_full_group_details, check_join_group, check_link_join, add_user_group,
get_short_group_details, calculate_balance, create_group_invite_service, ) from app.core.exceptions import (
GroupFullDetailsError, GroupNotFoundError, GroupCheckLinkJoinError, GroupAddUserError,
GroupShortDetailsError, GroupCalculateBalanceError, GroupInviteLinkCreateError, )

```

```

def test_get_full_group_details_success(db_session): user = User(name="Greg", email="greg@example.com",
pw="hashed") group = Group(name="Weekend", pw="pw", emoji=" 🌈 ") db_session.add_all([user, group])
db_session.flush()

```

```

membership = GroupMembers(user_id=user.id, group_id=group.id)
db_session.add(membership)

expense = Expense(
    amount=100.0,
    description="Hotel",
    group_id=group.id,
    paid_by_id=user.id,
    created_by_id=user.id,
)
db_session.add(expense)

```

```
db_session.flush()

split = ExpenseSplit(expense_id=expense.id, user_id=user.id, amount=100.0)
db_session.add(split)
db_session.flush()

result = get_full_group_details(group_id=group.id, db=db_session)

assert result.id == group.id
assert len(result.members) == 1
assert len(result.expenses) == 1
```

```
def test_get_full_group_details_failure(db_session, monkeypatch):
    def broken_query(*args, **kwargs):
        raise RuntimeError("boom")
```

```
    monkeypatch.setattr(db_session, "query", broken_query)

    with pytest.raises(GroupFullDetailsError):
        get_full_group_details(group_id=1, db=db_session)
```

```
def test_check_join_group_success(db_session):
    group = Group(name="Study", pw="joinme", emoji=None)
    db_session.add(group)
    db_session.flush()
```

```
    group_id = check_join_group(group_name="Study", group_pw="joinme",
                                db=db_session)

    assert group_id == group.id
```

```
def test_check_join_group_failure(db_session, monkeypatch):
    def broken_query(*args, **kwargs):
        raise RuntimeError("db down")
```

```
    monkeypatch.setattr(db_session, "query", broken_query)

    with pytest.raises(GroupNotFoundError):
        check_join_group(group_name="Nope", group_pw="invalid", db=db_session)
```

```
def test_check_link_join_success(db_session):
    creator = User(name="Hank", email="hank@example.com",
                    pw="hashed")
    group = Group(name="Picnic", pw="token", emoji=None)
    invite = GroupInvite(group_id=1,
                          created_by_id=1, token="abc123")
```

```
    db_session.add_all([creator, group])
    db_session.flush()

    invite.group_id = group.id
    invite.created_by_id = creator.id
    db_session.add(invite)
```

```
db_session.flush()

group_id = check_link_join(token_link="https://example.com/join?token=abc123",
db=db_session)

assert group_id == group.id
assert invite.used is True
```

```
def test_check_link_join_expired(db_session): creator = User(name="Ian", email="ian@example.com",
pw="hashed") group = Group(name="Retreat", pw="pw", emoji=None) invite = GroupInvite(group=group,
created_by=creator, token="expired") invite.used = True
```

```
db_session.add_all([creator, group, invite])
db_session.flush()

with pytest.raises(GroupCheckLinkJoinError):
    check_link_join(token_link="https://example.com/join?token=expired",
db=db_session)
```

```
def test_add_user_group_success(db_session): user = User(name="Jane", email="jane@example.com",
pw="hashed") group = Group(name="Brunch", pw="pw", emoji=None) db_session.add_all([user, group])
db_session.flush()
```

```
result = add_user_group(group_id=group.id, user=user, db=db_session)

assert any(member.id == user.id for member in result.members)
```

```
def test_add_user_group_failure(db_session, monkeypatch): user = User(name="Kyle",
email="kyle@example.com", pw="hashed") group = Group(name="Hike", pw="pw", emoji=None)
db_session.add_all([user, group]) db_session.flush()
```

```
def broken_flush(*args, **kwargs):
    raise RuntimeError("cannot flush")

monkeypatch.setattr(db_session, "flush", broken_flush)

with pytest.raises(GroupAddUserError): #not sure why this exception is not
being raised
    add_user_group(group_id=group.id, user=user, db=db_session)
```

```
def test_get_short_group_details_success(db_session): user = User(name="Liam", email="liam@example.com",
pw="hashed") group = Group(name="Cycling", pw="pw", emoji=None) db_session.add_all([user, group])
db_session.flush()
```

```
membership = GroupMembers(user_id=user.id, group_id=group.id)
db_session.add(membership)
db_session.flush()

groups = get_short_group_details(user_id=user.id, db=db_session)

assert len(groups) == 1
assert groups[0].id == group.id
```

```
def test_get_short_group_details_failure(db_session, monkeypatch):
    def broken_query(*args, **kwargs):
        raise RuntimeError("query failed")
```

```
monkeypatch.setattr(db_session, "query", broken_query)

with pytest.raises(GroupShortDetailsError):
    get_short_group_details(user_id=1, db=db_session)
```

```
def test_calculate_balance_success(db_session):
    user = User(name="Mia", email="mia@example.com",
pw="hashed")
    partner = User(name="Ned", email="ned@example.com", pw="hashed")
    group = Group(name="Ski", pw="pw", emoji=None)
    db_session.add_all([user, partner, group])
    db_session.flush()
```

```
expense = Expense(
    amount=120.0,
    description="Lift tickets",
    group_id=group.id,
    paid_by_id=user.id,
    created_by_id=user.id,
)
db_session.add(expense)
db_session.flush()

split_user = ExpenseSplit(expense_id=expense.id, user_id=user.id, amount=40.0)
split_partner = ExpenseSplit(expense_id=expense.id, user_id=partner.id,
amount=80.0)
db_session.add_all([split_user, split_partner])
db_session.flush()

balance = calculate_balance(user=user, group_id=group.id, db=db_session)

assert balance == pytest.approx(80.0)
```

```
def test_calculate_balance_failure(db_session, monkeypatch):
    def broken_query(*args, **kwargs):
        raise RuntimeError("sum failed")
```

```
monkeypatch.setattr(db_session, "query", broken_query)

with pytest.raises(GroupCalculateBalanceError):
```

```
calculate_balance(user=User(id=1, name="Temp", email="t@example.com",
pw="pw"), group_id=1, db=db_session)
```

```
def test_create_group_invite_service_success(db_session): user = User(name="Olive",
email="olive@example.com", pw="hashed") group = Group(name="Potluck", pw="pw", emoji=None)
db_session.add_all([user, group]) db_session.flush()
```

```
invite = create_group_invite_service(user_id=user.id, group_id=group.id,
db=db_session)

assert invite.group_id == group.id
assert invite.created_by_id == user.id
assert invite.token
```

```
def test_create_group_invite_service_failure(db_session, monkeypatch): user = User(name="Paul",
email="paul@example.com", pw="hashed") group = Group(name="Boating", pw="pw", emoji=None)
db_session.add_all([user, group]) db_session.flush()
```

```
def broken_flush(*args, **kwargs):
    raise RuntimeError("persist failed")

monkeypatch.setattr(db_session, "flush", broken_flush)

with pytest.raises(GroupInviteLinkCreateError): #not sure why this exception is
    not being raised
    create_group_invite_service(user_id=user.id, group_id=group.id,
db=db_session)
```

```
===== FILE: backend/tests/test_security.py ===== from datetime import datetime, timedelta, timezone
```

```
import pytest from fastapi import HTTPException from fastapi.security import HTTPAuthorizationCredentials from
jose import jwt
```

```
from app.core.config import settings from app.core.security import ( hash_password, verify_password,
create_access_token, decode_access_token, get_current_user, get_current_group, ) from app.db.models import
User, Group, GroupMembers
```

```
def test_hash_and_verify_password(): raw_password = "mysecretpassword" hashed =
hash_password(raw_password)
```

```
assert hashed != raw_password
assert verify_password(raw_password, hashed) is True
assert verify_password("wrong", hashed) is False
```

```
def test_create_and_decode_access_token(): token = create_access_token(user_id=123) payload =
decode_access_token(token)
```

```
assert payload["sub"] == "123"
```

```
def test_decode_expired_access_token(): expired_payload = {"sub": "1", "exp": datetime.now(timezone.utc) -  
timedelta(minutes=1), } expired_token = jwt.encode(expired_payload, settings.jwt_secret_key,  
algorithm=settings.jwt_algorithm)
```

```
with pytest.raises(HTTPException) as exc_info:  
    decode_access_token(expired_token)  
  
assert exc_info.value.status_code == 401  
assert exc_info.value.detail == "Token has expired"
```

```
def test_decode_invalid_access_token(): with pytest.raises(HTTPException) as exc_info:  
    decode_access_token("not-a-real-token")
```

```
assert exc_info.value.status_code == 401  
assert exc_info.value.detail == "Invalid token"
```

```
def test_get_current_user_success(db_session, auth_header): headers, user = auth_header creds =  
HTTPAuthorizationCredentials(scheme="Bearer", credentials=headers["Authorization"].split()[1])
```

```
current_user = get_current_user(creds=creds, db=db_session)  
  
assert current_user.id == user.id  
assert current_user.email == user.email
```

```
def test_get_current_user_missing_user(db_session): token = create_access_token(user_id=999) creds =  
HTTPAuthorizationCredentials(scheme="Bearer", credentials=token)
```

```
with pytest.raises(HTTPException) as exc_info:  
    get_current_user(creds=creds, db=db_session)  
  
assert exc_info.value.status_code == 403  
assert exc_info.value.detail == "User not found / doesn't exist"
```

```
def test_get_current_user_invalid_payload(db_session): token = jwt.encode({"exp": datetime.now(timezone.utc)  
+ timedelta(minutes=5)}, settings.jwt_secret_key, algorithm=settings.jwt_algorithm) creds =  
HTTPAuthorizationCredentials(scheme="Bearer", credentials=token)
```

```
with pytest.raises(HTTPException) as exc_info:  
    get_current_user(creds=creds, db=db_session)
```



```
assert exc_info.value.status_code == 401
assert exc_info.value.detail == "Invalid token payload"
```

```
def test_get_current_group_success(db_session, auth_header): headers, user = auth_header group =
Group(name="Trip", pw="secret", emoji=" 😊 ") db_session.add(group) db_session.flush()
```

```
membership = GroupMembers(user_id=user.id, group_id=group.id)
db_session.add(membership)
db_session.flush()

creds = HTTPAuthorizationCredentials(scheme="Bearer",
credentials=headers["Authorization"].split()[1])
current_user = get_current_user(creds=creds, db=db_session)

context = get_current_group(group_id=group.id, db=db_session,
current_user=current_user)

assert context.group.id == group.id
assert context.user.id == user.id
```

```
def test_get_current_group_forbidden(db_session, auth_header): headers, user = auth_header group =
Group(name="Secret", pw="hidden", emoji=None) db_session.add(group) db_session.flush()
```

```
creds = HTTPAuthorizationCredentials(scheme="Bearer",
credentials=headers["Authorization"].split()[1])
current_user = get_current_user(creds=creds, db=db_session)

with pytest.raises(HTTPException) as exc_info:
    get_current_group(group_id=group.id, db=db_session,
current_user=current_user)

assert exc_info.value.status_code == 403
assert exc_info.value.detail == "User does not have access to this group"
```

```
def test_get_current_group_invalid_token(db_session): user = User(name="Orphan",
email="nope@example.com", pw=hash_password("pwd")) db_session.add(user) db_session.flush()
```

```
group = Group(name="OrphanGroup", pw="pw", emoji=None)
db_session.add(group)
db_session.flush()

with pytest.raises(HTTPException):
    get_current_group(group_id=group.id, db=db_session, current_user=user)
```

===== FILE: backend/app/init.py =====

===== FILE: backend/app/api/init.py =====

```
===== FILE: backend/app/api/auth.py ===== from fastapi import APIRouter, Depends, HTTPException, status
from sqlalchemy.orm import Session from logging import Logger
```

```
from app.db.session import get_db from app.db.schemas import UserCreate, AuthOut, UserLogin from
app.db.models import User from app.core.exceptions import AuthJwtCreationError, AuthCredentialsError from
app.core.security import hash_password, verify_password, create_access_token from app.core.logger import
get_request_logger
```

```
router = APIRouter()
```

```
#signup @router.post("/signup", response_model=AuthOut) def signup( user: UserCreate, db: Session =
Depends(get_db), logger: Logger = Depends(get_request_logger), ): try: hashed_pw = hash_password(user.pw)
new_user = User(name=user.name, email=user.email, pw=hashed_pw) db.add(new_user) db.commit()
db.refresh(new_user)
```

```
    if not new_user:
        logger.error("user signup failed", extra={"email": user.email})
        raise AuthCredentialsError

    token = create_access_token(user_id=new_user.id)
    if not token:
        raise AuthJwtCreationError

    logger.info("user signup", extra={"user_id": new_user.id})
    return {"access_token": token, "user": new_user} #AuthOut
except AuthCredentialsError:
    db.rollback()
    raise HTTPException(
        status_code=status.HTTP_400_BAD_REQUEST,
        detail="Invalid email or password"
    )
except AuthJwtCreationError:
    db.rollback()
    raise HTTPException(
        status_code=status.HTTP_500_INTERNAL_SERVER_ERROR,
        detail="Error creating user token"
    )
except Exception as e:
    db.rollback()
    logger.error(f"error in create group endpoint: {e}")
    raise HTTPException(status_code=500, detail="Unexpected server error")
```

```
#login @router.post("/login", response_model=AuthOut) def login( user: UserLogin, db: Session =
Depends(get_db), logger: Logger = Depends(get_request_logger), ): try: logged_user =
db.query(User).filter(User.email == user.email).first()
```

```
    if not logged_user or not verify_password(user.pw, logged_user.pw): #hashed
password second
        logger.warning("user login failed", extra={"email": user.email})
        raise AuthCredentialsError
    token = create_access_token(user_id=logged_user.id)
```

```

    if not token:
        raise AuthJwtCreationError
    logger.info("user login", extra={"user_id": logged_user.id})

    return {"access_token": token, "user": logged_user} #AuthOut

except AuthCredentialsError:
    raise HTTPException(
        status_code=status.HTTP_400_BAD_REQUEST,
        detail="Invalid email or password"
    )
except AuthJwtCreationError:
    raise HTTPException(
        status_code=status.HTTP_500_INTERNAL_SERVER_ERROR,
        detail="Error creating user token"
    )
except Exception as e:
    logger.error(f"error in create group endpoint: {e}")
    raise HTTPException(status_code=500, detail="Unexpected server error")

```

===== FILE: backend/app/api/expenses.py ===== #translates pure http --> expenses class

```

from fastapi import APIRouter, Depends, HTTPException, status from sqlalchemy.orm import Session #type:
ignore from logging import Logger

```

```

from app.db.session import get_db from app.db.schemas import ExpenseCreate, ExpenseOut, ExpenseUpdate,
ExpenseDelete from app.db.models import Expense from app.core.logger import get_request_logger from
app.core.exceptions import ExpenseCreationError, ExpenseEditError, ExpenseNotFoundError

```

```

from app.services.expense_service import create_expense_service, edit_expense_service from app.core.security
import get_current_group, GroupContext

```

```

router = APIRouter()

```

```

@router.post("/{group_id}/create-expense", response_model=ExpenseOut) def create_expense( group_id: int,
#safer to include it in url expense: ExpenseCreate, db: Session = Depends(get_db), ctx: GroupContext =
Depends(get_current_group), #to make sure that this user is a member of the curent group logger: Logger =
Depends(get_request_logger), ): try: logger.info("expense create payload received", extra={"group_id":
ctx.group.id, "paid_by": expense.paid_by_id, "created_by": ctx.user.id}) new_expense =
create_expense_service(new_expense=expense, user_id=ctx.user.id, group_id=ctx.group.id, db=db) db.commit()
logger.debug("expense created", extra={"expense_id": new_expense.id})

```

```

    return new_expense

except ExpenseCreationError:
    db.rollback()
    raise HTTPException(
        status_code=status.HTTP_400_BAD_REQUEST,
        detail="Error creating expense"
    )
except Exception as e:
    logger.error(f"Error creating expense: {e}")

```

```
db.rollback()
raise HTTPException(status_code=500, detail="Unexpected server error")
```

```
@router.post("/{group_id}/edit-expense", response_model=ExpenseOut) def edit_expense( group_id: int,
expense: ExpenseUpdate, db: Session = Depends(get_db), ctx: GroupContext = Depends(get_current_group),
logger: Logger = Depends(get_request_logger), ): try: logger.info("expense edit payload recieved", extra=
{"group_id": ctx.group.id, "user_id": ctx.user.id, "expense_id": expense.id}) updated_expense =
edit_expense_service(expense_update=expense, user_id=ctx.user.id, group_id=ctx.group.id, db=db)
db.add(updated_expense) db.commit() db.refresh(updated_expense) logger.debug("expense updated", extra=
{"expense_id": updated_expense.id})
```

```
        return updated_expense

    except ExpenseNotFoundError:
        db.rollback()
        raise HTTPException(
            status_code=status.HTTP_404_NOT_FOUND,
            detail="Expense not found"
        )
    except ExpenseEditError:
        db.rollback()
        raise HTTPException(
            status_code=status.HTTP_500_INTERNAL_SERVER_ERROR,
            detail="Error creating expense"
        )
    except Exception as e:
        logger.error(f"Error editing expense: {e}")
        db.rollback()
        raise HTTPException(status_code=500, detail="Unexpected server error")
```

```
@router.post("/{group_id}/delete-expense") def delete_expense( group_id: int, expense: ExpenseDelete, db:
Session = Depends(get_db), ctx: GroupContext = Depends(get_current_group), logger: Logger =
Depends(get_request_logger), ): try: logger.info("expense delete payload recieved", extra={"group_id":
ctx.group.id, "user_id": ctx.user.id, "expense_id": expense.id})
expense_to_delete = ( db.query(Expense) .filter(Expense.id == expense.id, Expense.group_id == group_id) .first()
) if not expense_to_delete: raise ExpenseNotFoundError db.delete(expense_to_delete) db.commit()
```

```
        return {"msg": "Expense deleted"}

    except ExpenseNotFoundError:
        db.rollback()
        raise HTTPException(
            status_code=status.HTTP_404_NOT_FOUND,
            detail="Expense not found"
        )
    except Exception as e:
        logger.error(f"Error deleting expense: {e}")
        db.rollback()
        raise HTTPException(status_code=500, detail="Unexpected server error")
```

```
===== FILE: backend/app/api/groups.py ===== #translates pure http --> group class
```

```
from fastapi import APIRouter, Depends, HTTPException, status from sqlalchemy.orm import Session from logging
import Logger from typing import List from datetime import datetime, timedelta, timezone
```

```
from app.db.session import get_db from app.db.schemas import GroupCreate, GroupJoinIn, GroupOut,
GroupShortOut, GroupInviteOut from app.db.models import Group, User, GroupMembers from
app.services.group_service import get_full_group_details, check_join_group, check_link_join,
get_short_group_details, calculate_balance, add_user_group, create_group_invite_service from
app.core.exceptions import ( GroupFullDetailsError, GroupCheckPwJoinError, GroupCheckLinkJoinError,
GroupAddUserError, GroupShortDetailsError, GroupInviteLinkCreateError, GroupNotFoundError,
GroupUserAlreadyJoinedError ) from app.core.security import get_current_user, get_current_group,
GroupContext from app.core.logger import get_request_logger
```

```
router = APIRouter()
```

```
@router.post("/create", response_model=GroupOut) def create_group( group: GroupCreate, db: Session =
Depends(get_db), current_user: User = Depends(get_current_user), #from jwt logger: Logger =
Depends(get_request_logger), ): try: logger.debug("group create payload received", extra={"group": group.dict(),
"user_id": current_user.id}) new_group = Group(pw=group.group_pw, name=group.name, emoji=group.emoji)
```

```
    #ensure the creator is attached through the association table
    new_member = GroupMembers(user=current_user, group=new_group)
    new_group.member_associations.append(new_member)

    db.add(new_group)
    db.commit()
    db.refresh(new_group)

    logger.info("group created", extra={"group_id": new_group.id})

    group_details = get_full_group_details(new_group.id, db=db)
    #calc balances
    for member in group_details.members:
        member.balance = calculate_balance(user=member,
group_id=group_details.id, db=db)

    return group_details

except GroupFullDetailsError:
    db.rollback()
    raise HTTPException(
        status_code=status.HTTP_500_INTERNAL_SERVER_ERROR,
        detail="Error polling db for group details"
    )
except GroupNotFoundError:
    db.rollback()
    raise HTTPException(
        status_code=status.HTTP_404_NOT_FOUND,
        detail="Group not found"
    )
```

```

except Exception as e:
    db.rollback()
    logger.error(f"error in create group endpoint: {e}")
    raise HTTPException(status_code=500, detail="Unexpected server error")

```

```

@router.post("/join", response_model=GroupOut) def join_group( group: GroupJoinIn, db: Session =
Depends(get_db), current_user: User = Depends(get_current_user), logger: Logger =
Depends(get_request_logger), ): try: logger.debug("join group attempt", extra={"type": group.pw_auth if
group.pw_auth else group.link_auth}) group_id = check_join_group(group_name=group.pw_auth.group_name,
group_pw=group.pw_auth.group_pw, db=db) if group.pw_auth else check_link_join(token_link=group.link_auth,
db=db) #group not found error raised instead of checking group_id val joined_group_details =
add_user_group(group_id=group_id, user=current_user, db=db) db.commit()

```

```

        for member in joined_group_details.members:
            member.balance = calculate_balance(user=member,
group_id=joined_group_details.id, db=db)
            #this is only querying, no db commit needed
        return joined_group_details

except GroupNotFoundError:
    db.rollback()
    raise HTTPException(
        status_code=status.HTTP_404_NOT_FOUND,
        detail="Group not found"
    )
except GroupAddUserError:
    db.rollback()
    raise HTTPException(
        status_code=status.HTTP_500_INTERNAL_SERVER_ERROR,
        detail="Error adding user to group relationship"
    )
except GroupUserAlreadyJoinedError:
    db.rollback()
    raise HTTPException(
        status_code=status.HTTP_400_BAD_REQUEST,
        detail="You already are in this group"
    )
except GroupCheckLinkJoinError:
    db.rollback()
    raise HTTPException(
        status_code=status.HTTP_400_BAD_REQUEST,
        detail="Invite link has already been used or is expired. Request
another"
    )
except GroupCheckPwJoinError:
    db.rollback()
    raise HTTPException(
        status_code=status.HTTP_400_BAD_REQUEST,
        detail="Incorrect password or name"
    )
except Exception as e:

```

```

db.rollback()
logger.error(f"error in join group endpoint: {e}")
raise HTTPException(status_code=500, detail="Unexpected server error")

```

```

@router.get("/view-short", response_model=List[GroupShortOut]) def view_all_groups( current_user: User =
Depends(get_current_user), db: Session = Depends(get_db), logger: Logger = Depends(get_request_logger), ):
try: logger.debug("view all groups attempt", extra={"User id": current_user.id, "User name ": current_user.name})
return get_short_group_details(user_id=current_user.id, db=db) except GroupShortDetailsError: raise
HTTPException( status_code=status.HTTP_404_INTERNAL_ERROR, detail="Error finding user's groups" ) except
Exception as e: #no db rollback cause only reading raise HTTPException(status_code=500, detail="Unexpected
server error")

```

```

@router.get("/{group_id}", response_model=GroupOut) def view_group( ctx: GroupContext =
Depends(get_current_group), db: Session = Depends(get_db), logger: Logger = Depends(get_request_logger), ):
try: logger.debug("view group attempt", extra={"group_name": ctx.group.name, "user_name": ctx.user.name})

```

```

        joined_group_details = get_full_group_details(ctx.group.id, db=db)
        for member in joined_group_details.members:
            member.balance = calculate_balance(user=member,
group_id=joined_group_details.id, db=db)
        return joined_group_details
except GroupFullDetailsError:
    db.rollback()
    raise HTTPException(
        status_code=status.HTTP_500_INTERNAL_SERVER_ERROR,
        detail="Error polling db for group details"
    )
except Exception as e:
    logger.error(f"error in view group endpoint: {e}")
    raise HTTPException(status_code=500, detail="Unexpected server error")

```

```

@router.get("/{group_id}/create-invite", response_model=GroupInviteOut) def create_group_invite( ctx:
GroupContext = Depends(get_current_group), db: Session = Depends(get_db), logger: Logger =
Depends(get_request_logger) ): try: now = datetime.now(timezone.utc) expires_at = now +
timedelta(minutes=10) #10 min token expiry time inv = create_group_invite_service(user_id=ctx.user.id,
group_id=ctx.group.id, db=db, expires_at=expires_at) db.commit() return inv

```

```

except GroupInviteLinkCreateError:
    db.rollback()
    raise HTTPException(
        status_code=status.HTTP_500_INTERNAL_SERVER_ERROR,
        detail="Error creating group invite"
    )
except Exception as e:
    db.rollback()
    logger.error(f"Error creating group invite: {e}")
    raise HTTPException(status_code=500, detail="Unexpected server error")

```

```
===== FILE: backend/app/core/config.py ===== #basic settings; db url, env vars from pydantic_settings import
BaseSettings
```

```
class Settings(BaseSettings): database_user: str database_pw: str database_name: str jwt_secret_key: str
jwt_algorithm: str jwt_expiration_minutes: int
```

```
log_format: str
base_logger_name: str

@property
def database_url(self):
    return f"postgresql+psycpg2://{self.database_user}:
{self.database_pw}@db:5432/{self.database_name}"

class Config: #tell pydantic to load variables from .env
    env_file = ".env"
```

```
settings = Settings()
```

```
===== FILE: backend/app/core/exceptions.py ===== #defining custom exception models for better readability and
making services framework (fastapi) agnostic
```

```
*****
***** #EXPENSES
*****
***** class ExpenseCreationError(Exception):
"""Generic expense creation error""" pass

class ExpenseEditError(Exception): """Generic expense edit error""" pass

class ExpenseNotFoundError(Exception): """Generic fallback""" pass

*****
***** #GROUPS
*****
***** class GroupFullDetailsError(Exception):
"When get_full_group_details service fails" pass

class GroupCalculateBalanceError(Exception): """When calculate_balance service fails""" pass

class GroupCheckPwJoinError(Exception): """When check_join_group (pw auth) service fails""" pass

class GroupCheckLinkJoinError(Exception): """When check_link_join (link auth) service fails""" pass

class GroupAddUserError(Exception): """When add_user_group service fails""" pass

class GroupUserAlreadyJoinedError(Exception): """When a user already exists in the group and the db raises an
integrety error (due to PK constraint on group_id and user_id)""" pass

class GroupShortDetailsError(Exception): """When get_short_group_details service fails""" pass

class GroupInviteLinkCreateError(Exception): """When create_group_invite_service service fails""" pass
```



```
#generic/reusable class GroupNotFoundError(Exception): """generic error msg for invalid inputs"""

#*****
***** #AUTH
#*****
***** class AuthJwtCreationError(Exception):
""" Important exception for auth to check jwt creation worked. Dont want to pass an empty token to frontend as
app relies on it """ pass

class AuthCredentialsError(Exception): """Generic credentials check""" pass

===== FILE: backend/app/core/logger.py ===== import logging from logging import Logger from typing import
Optional from fastapi import Request

from app.core.config import settings

def setup_logging(level: int = logging.INFO) -> Logger: logger = logging.getLogger(settings.base_logger_name) if
logger.handlers: return logger
```

```
logger.setLevel(level)
stream_handler = logging.StreamHandler()
stream_handler.setLevel(level)
stream_handler.setFormatter(logging.Formatter(settings.log_format))
logger.addHandler(stream_handler)
logger.propagate = False

return logger
```

```
def get_request_logger(request: Request, name: Optional[str] = None) -> Logger: base_logger: Logger =
request.app.state.logger if name: return base_logger.getChild(name)
```

```
endpoint = request.scope.get("endpoint")
if endpoint:
    return base_logger.getChild(endpoint.__qualname__)

return base_logger.getChild(request.scope.get("path", "unknown"))
```

```
def get_module_logger(name: str) -> Logger: base_logger = logging.getLogger(settings.base_logger_name)
```

```
return base_logger.getChild(name)
```

```
===== FILE: backend/app/core/security.py ===== #for later; jwt + pw hashing from fastapi import HTTPException,
Depends, status from passlib.context import CryptContext from datetime import datetime, timezone, timedelta
from fastapi.security import HTTPBearer, HTTPAuthorizationCredentials from sqlalchemy.orm import Session from
jose import jwt, ExpiredSignatureError, JWTError from dataclasses import dataclass

from app.core.config import settings from app.db.session import get_db from app.db.models import User, Group,
GroupMembers
```

```
#inits pwd_context = CryptContext(schemes=["argon2", "bcrypt"], deprecated="auto") bearer_scheme = HTTPBearer()

@dataclass class GroupContext: group: Group user: User

#basic hashing def hash_password(password: str): return pwd_context.hash(password)

def verify_password(password: str, hashed_password: str): return pwd_context.verify(password, hashed_password)

#jwt def create_access_token(user_id: str | int): expire_d = settings.jwt_expiration_minutes now = datetime.now(timezone.utc) expire = now + timedelta(minutes=expire_d) to_encode = {"exp": expire, "sub": str(user_id)} #always gonna use user_id for subject return jwt.encode(to_encode, settings.jwt_secret_key, algorithm=settings.jwt_algorithm)

def decode_access_token(token: str): try: return jwt.decode(token, settings.jwt_secret_key, algorithms=[settings.jwt_algorithm], options={"verify_exp": True}) except ExpiredSignatureError: raise HTTPException(status_code=401, detail="Token has expired") except JWTErrors: raise HTTPException(status_code=401, detail="Invalid token")

#dependency for jwt def get_current_user( creds: HTTPAuthorizationCredentials = Depends(bearer_scheme), db: Session = Depends(get_db), ) -> User: """ Because this is a dependency it needs a HTTPException rather than a custom exception. This ensures fastapi stops the request immediatley and endpoints dont get called """ payload = decode_access_token(creds.credentials) user_id = payload.get("sub") if not user_id: raise HTTPException(status_code=401, detail="Invalid token payload")
```

```
user = db.query(User).get(int(user_id)) #return matching user
if not user:
    raise HTTPException(status_code=403, detail="User not found / doesn't exist")

return user #since this is for internal use only, fine to return entire user object
```

```
#dependency using jwt and parameter group id def get_current_group( group_id: int, db: Session = Depends(get_db), current_user: User = Depends(get_current_user), ) -> GroupContext: #verifying that User is part of Group """ Because this is a dependency it needs a HTTPException rather than a custom exception. This ensures fastapi stops the request immediatley and endpoints dont get called """
```

```
group = (
    db.query(Group)
    .join(Group.member_associations) #join groups to group_members
    .filter(
        Group.id == group_id,
        GroupMembers.user_id == current_user.id
    )
    .first()
)

if not group:
    raise HTTPException(
```

```

        status_code=status.HTTP_403_FORBIDDEN,
        detail="User does not have access to this group",
    )

    return GroupContext(group=group, user=current_user)

```

===== FILE: backend/app/db/base.py ===== #define base from sqlalchemy.orm import declarative_base

```
Base = declarative_base()
```

===== FILE: backend/app/db/models.py ===== #sqlalchemy ORM models import secrets

```
from .base import Base
from sqlalchemy import Column, Integer, Text, String, ForeignKey, Boolean, Float,
DateTime, func
from sqlalchemy.orm import relationship
```

```
class User(Base):
    tablename = "user"
```

```

    id = Column(Integer, autoincrement=True, primary_key=True)
    name = Column(String, nullable=False)
    pw = Column(String, nullable=False)
    email = Column(String, nullable=False)

    group_associations = relationship("GroupMembers", back_populates="user")
    groups = relationship("Group", secondary="group_members", viewonly=True)

    expenses_paid = relationship("Expense", foreign_keys="Expense.paid_by_id",
                                back_populates="paid_by")
    expenses_created = relationship("Expense",
                                    foreign_keys="Expense.created_by_id", back_populates="created_by")
    created_invites = relationship("GroupInvite", back_populates="created_by")

```

```
class Group(Base):
    tablename = "group"
```

```

    id = Column(Integer, autoincrement=True, primary_key=True)
    name = Column(String, nullable=False)
    pw = Column(String, nullable=False)

    emoji = Column(Text) #emoji code; the icon representing group

    member_associations = relationship("GroupMembers", back_populates="group")
    members = relationship("User", secondary="group_members", viewonly=True)
    expenses = relationship("Expense", back_populates="group")
    invites = relationship("GroupInvite", back_populates="group", cascade="all,
delete-orphan")

```

```
class GroupMembers(Base):
    tablename = "group_members"
```

```

    user_id = Column(Integer, ForeignKey("user.id"), primary_key=True)
    group_id = Column(Integer, ForeignKey("group.id"), primary_key=True)

```

```

user = relationship("User", back_populates="group_associations")
group = relationship("Group", back_populates="member_associations")

```

```
class Expense(Base): tablename = "expense"
```

```

id = Column(Integer, autoincrement=True, primary_key=True)
amount = Column(Float, nullable=False)
description = Column(String)

group_id = Column(Integer, ForeignKey("group.id"), nullable=False)
paid_by_id = Column(Integer, ForeignKey("user.id"), nullable=False)
created_by_id = Column(Integer, ForeignKey("user.id"), nullable=False)

group = relationship("Group", back_populates="expenses")
paid_by = relationship("User", foreign_keys=[paid_by_id],
back_populates="expenses_paid")
created_by = relationship("User", foreign_keys=[created_by_id],
back_populates="expenses_created")

splits = relationship("ExpenseSplit", back_populates="expense", cascade="all,
delete-orphan", single_parent=True)

```

```

class ExpenseSplit(Base): tablename = "expense_split" id = Column(Integer, autoincrement=True,
primary_key=True) expense_id = Column(Integer, ForeignKey("expense.id"), nullable=False) user_id =
Column(Integer, ForeignKey("user.id"), nullable=False)

```

```

amount = Column(Float, nullable=False)

expense = relationship("Expense", back_populates="splits")
user = relationship("User")

```

```
class GroupInvite(Base): tablename = "group_invite"
```

```

id = Column(Integer, autoincrement=True, primary_key=True)
token = Column(String, unique=True, nullable=False, index=True, default=lambda:
secrets.token_urlsafe(16))
group_id = Column(Integer, ForeignKey("group.id"), nullable=False)
created_by_id = Column(Integer, ForeignKey("user.id"), nullable=False)

expires_at = Column(DateTime(timezone=True), nullable=True)
used = Column(Boolean, default=False, nullable=False)
created_at = Column(DateTime(timezone=True), server_default=func.now(),
nullable=False)

group = relationship("Group", back_populates="invites")
created_by = relationship("User", back_populates="created_invites")

```

```

===== FILE: backend/app/db/schemas.py ===== #pydantic models, validates request and response bodies from
pydantic import BaseModel from typing import Optional, List

#new user signup; no reuse class UserCreate(BaseModel): name: str pw: str email: str

class UserLogin(BaseModel): email: str pw: str

#generic model for inputting a user (user_id) class UserIn(BaseModel): id: int name: str

class UserOut(BaseModel): id: int name: str #limit this info as it gets re-used for public stuff class Config:
orm_mode = True

class AuthOut(BaseModel): access_token: str token_type: str = "bearer" user: UserOut

class ExpenseSplitIn(BaseModel): user: UserIn amount: float

#new expense class ExpenseCreate(BaseModel): paid_by_id: int #can be any user id, so dont rely on jwt amount:
float description: Optional[str] splits: List[ExpenseSplitIn]

class ExpenseUpdate(BaseModel): id: int expense: ExpenseCreate

class ExpenseDelete(BaseModel): id: int

#encapsulate splits in the expense_split table class ExpenseSplitOut(BaseModel): user: UserOut amount: float

```

```

class Config:
    orm_mode = True

```

```

#list all splits for one expense class ExpenseOut(BaseModel): id: int amount: float description: Optional[str]

```

```

    paid_by: UserOut
    splits: List[ExpenseSplitOut]

class Config:
    orm_mode = True

```

```

class ExpenseIn(BaseModel): id: int amount: Optional[float] = None description: Optional[str] = None paid_by_id:
Optional[int] = None splits: Optional[List[ExpenseSplitIn]] = None

```

```

#creating a new group class GroupCreate(BaseModel): #user_id: int no longer needed cause jwt name: str
group_pw: str emoji: Optional[str]

```

```

#joining an existing group (for now just name + pw, later we can do a link or smth) class
PasswordAuth(BaseModel): group_name: str group_pw: str

```

```

class GroupJoinIn(BaseModel): pw_auth: Optional[PasswordAuth] = None link_auth: Optional[str] = None

```

```

class UserBalanceOut(BaseModel): id: int name: str balance: float

```

```

#info received when you click on an actual group class GroupOut(BaseModel): id: int name: str emoji: Optional[str]
members: List[UserBalanceOut] expenses: List[ExpenseOut]

```

```
class Config:
    orm_mode = True
```

```
class GroupShortOut(BaseModel): id: int name: str emoji: Optional[str]
```

```
class GroupInviteOut(BaseModel): token: str
```

```
===== FILE: backend/app/db/session.py ===== #DBSession object; manages connections import time from
sqlalchemy import create_engine, text from sqlalchemy.orm import sessionmaker from sqlalchemy.exc import
OperationalError
```

```
from app.db.base import Base from app.core.config import settings from app.db import models #to create tables
if neccecesary from logging import Logger
```

```
MAX_RETRIES=10 WAIT_SECONDS=2
```

```
for attempt in range(1, MAX_RETRIES + 1): try: #logger.info(f"Attempt {attempt}/{MAX_RETRIES}: connecting to
database...") engine = create_engine(settings.database_url, echo=False) #Test connection with engine.connect()
as conn: conn.execute(text("SELECT 1")) #logger.info("✅ Database connection successful.") break except
OperationalError as e: #logger.warning(f"❌ Database not ready ({e}); retrying in {WAIT_SECONDS}s...")
time.sleep(WAIT_SECONDS) else: raise RuntimeError(f"❌ Could not connect to the database after
{MAX_RETRIES} attempts.")
```

```
#engine = create_engine(settings.database_url, echo=False) Base.metadata.create_all(engine) #create tables if it
doesn't exist
```

```
SessionLocal = sessionmaker(autocommit=False, autoflush=False, bind=engine)
```

```
#for fastapi injection def get_db(): db = SessionLocal() try: yield db finally: db.close()
```

```
===== FILE: backend/app/main.py ===== from fastapi import FastAPI, Request
```

```
from fastapi.responses import JSONResponse from fastapi.middleware.cors import CORSMiddleware from
sqlalchemy.exc import IntegrityError
```

```
from app.core.logger import setup_logging from app.api import groups, expenses, auth
```

```
app_logger = setup_logging()
```

```
app = FastAPI() app.state.logger = app_logger
```

```
origins = [
```

```
"http://localhost:3000",
"http://127.0.0.1:3000"
```

```
]
```

```
app.add_middleware( CORSMiddleware, allow_origins=origins, allow_credentials=True, allow_methods=[" "],
allow_headers=[""], )
```

```
app.include_router(groups.router, prefix="/groups", tags=["groups"]) app.include_router(expenses.router,
prefix="/expenses", tags=["expenses"]) app.include_router(auth.router, prefix="/auth", tags=["auth"])
```

#global exception handler; any IntegrityError (data/database) will only raise this response

```
@app.exception_handler(IntegrityError) def integrity_error_handler(request: Request, exc: IntegrityError): return
JSONResponse( status_code=400, content={"detail": "Database integrity error"} )
```

```
===== FILE: backend/app/services/expense_service.py ===== from sqlalchemy.orm import Session from fastapi
import Depends
```

```
from app.db.models import Group, Expense, ExpenseSplit, User from app.db.schemas import ExpenseCreate,
ExpenseOut, ExpenseIn, ExpenseUpdate from app.core.exceptions import ExpenseCreationError,
ExpenseEditError, ExpenseNotFoundError from app.db.session import get_db from app.core.logger import
get_module_logger
```

```
logger = get_module_logger(name)
```

```
def create_expense_service( new_expense: ExpenseCreate, user_id: int, group_id: int, db: Session =
Depends(get_db) ) -> Expense: logger.debug( "creating expense", extra={"group_id": group_id, "created by":
user_id, "paid_by_id": new_expense.paid_by_id}, )
```

```
try:
    expense = Expense(
        amount=new_expense.amount,
        description=new_expense.description,
        paid_by_id=new_expense.paid_by_id,
        group_id=group_id, #already checked and found in security.py
        created_by_id=user_id,
    )

    #iterate over splits
    for split in new_expense.splits:
        expense.splits.append(
            ExpenseSplit(user_id=split.user.id, amount=split.amount)
        )

    db.add(expense)
    db.flush()
    db.refresh(expense)
    logger.info("expense object created")

    return expense

except Exception as e:
    logger.error(
        "expense object creation failed",
        extra={"group_id": group_id},
    )
    raise ExpenseCreationError from e
```

```
def edit_expense_service( expense_update: ExpenseUpdate, user_id: int, group_id: int, db: Session, ) -> Expense:
logger.debug( "editing expense", extra={"group_id": group_id, "edited_by": user_id, "expense_id":
```

```
expense_update.id}, )
```

```
try:
    expense = (
        db.query(Expense)
        .filter(Expense.id == expense_update.id, Expense.group_id == group_id)
        .first()
    )

    if not expense:
        raise ExpenseNotFoundError

    edited_expense = expense_update.expense
    expense.amount = edited_expense.amount
    expense.description = edited_expense.description
    expense.paid_by_id = edited_expense.paid_by_id

    if edited_expense.splits is not None:
        expense.splits.clear()
        db.flush()

        for split in edited_expense.splits:
            expense.splits.append(
                ExpenseSplit(user_id=split.user.id, amount=split.amount)
            )
        db.flush()
        logger.info("expense updated", extra={"expense_id": expense_update.id})
        return expense

except Exception as e:
    logger.exception(
        "expense edit failed",
        extra={"group_id": group_id, "expense_id": expense_update.id},
    )
    raise ExpenseEditError from e
```

```
===== FILE: backend/app/services/group_service.py ===== #logic for creating groups; eg. only unique users per
group, max amount, etc. import secrets from datetime import datetime, timezone
```

```
from sqlalchemy.orm import Session, joinedload from sqlalchemy.exc import IntegrityError from sqlalchemy import
func from urllib.parse import urlparse, parse_qs
```

```
from app.db.models import Group, Expense, ExpenseSplit, User, GroupMembers, GroupInvite from
app.db.schemas import GroupOut, GroupShortOut, GroupInviteOut from app.core.exceptions import (
    GroupFullDetailsError, GroupCalculateBalanceError, GroupCheckPwJoinError, GroupCheckLinkJoinError,
    GroupAddUserError, GroupShortDetailsError, GroupInviteLinkCreateError, GroupNotFoundError,
    GroupUserAlreadyJoinedError ) from app.core.logger import get_module_logger
```

```
logger = get_module_logger(name)
```



```
def get_full_group_details(group_id: int, db: Session) -> GroupOut: try: group_details = ( db.query(Group)
.options( joinedload(Group.members), joinedload(Group.expenses) #.joinedload(Expense.paid_by)
.joinedload(Expense.splits) .joinedload(ExpenseSplit.user)
) .filter(Group.id == group_id) .first() )
```

```
    logger.debug("group details loaded", extra={"group_id": group_id})

    if not group_details:
        logger.warning("group lookup failed", extra={"group_id": group_id})
        raise GroupNotFoundError from e

    return group_details
except Exception as e:
    logger.error(f"Error loading group details: {e}")
    raise GroupFullDetailsError from e
```

```
def check_join_group(group_name: str, group_pw: str, db: Session) -> int: try: group = ( db.query(Group)
.filter(Group.name == group_name) .filter(Group.pw == group_pw) .first() )
```

```
    if not group:
        logger.warning("group lookup failed", extra={"group_name": group_name})
        raise GroupCheckPwJoinError from e
    return group.id

except Exception as e:
    logger.error(f"Error checking join group: {e}")
    raise GroupNotFoundError from e
```

```
def check_link_join(token_link: str, db: Session) -> int: try: parsed = urlparse(token_link) query_params =
parse_qs(parsed.query) token = query_params.get("token", [None])[0]
```

```
    logger.info(f"Token to check: {token}")

    invite = (
        db.query(GroupInvite)
        .filter(GroupInvite.token == token)
        .first()
    )

    if not invite or invite.used or (invite.expires_at and invite.expires_at <
datetime.now(timezone.utc)):
        raise GroupCheckLinkJoinError

    #mark as used
    invite.used = True
    db.add(invite)
    db.flush()
    db.refresh(invite)
```

```

        return invite.group_id

    except Exception as e:
        db.rollback() #cause we change used field
        logger.error(f"Error checking link join: {e}")
        raise GroupCheckLinkJoinError from e

```

```

def add_user_group(group_id: int, user: User, db: Session) -> GroupOut:
    try:
        group = ( db.query(Group)
        .filter(Group.id == group_id) .first() )
        new_member = GroupMembers(user=user, group=group)
        group.member_associations.append(new_member)

```

```

        db.add(new_member)
        db.flush() #best practice; only commit and rollback endpoint as it owns
        request lifecycle
        db.refresh(new_member)
        return get_full_group_details(group_id=group_id, db=db)
    except IntegrityError:
        logger.error("User already added to group")
        raise GroupUserAlreadyJoinedError
    except Exception as e:
        logger.error(f"Error adding user to group: {e}")
        raise GroupAddUserError from e

```

```

def get_short_group_details(user_id: int, db: Session) -> list[GroupShortOut]:
    try:
        logger.debug("group list request received")
        group_list = ( db.query(Group) .join(Group.members) .filter(User.id == user_id) .all() )
        logger.debug("group short list loaded", extra={"User_id": user_id})
        return group_list
    except Exception as e:
        logger.error(f"Error getting short group list: {e}")
        raise GroupShortDetailsError from e

```

```

def calculate_balance(user: User, group_id: int, db: Session):
    try:
        #total paid by user_id in this group
        total_paid = (
        db.query(func.coalesce(func.sum(Expense.amount), 0.0))
        .filter(Expense.group_id == group_id,
        Expense.paid_by_id == user.id) .scalar() )

```

```

        #total owed by the user in this group (splits)
        total_owed = (
            db.query(func.coalesce(func.sum(ExpenseSplit.amount), 0.0))
            .join(Expense)
            .filter(Expense.group_id == group_id, ExpenseSplit.user_id == user.id)
            .scalar()
        )

        balance = float(total_paid) - float(total_owed)
        return balance

    except Exception as e:
        logger.error(f"Error in calculate balance service: {e}")
        raise GroupCalculateBalanceError from e

```

```
def create_group_invite_service(user_id: int, group_id: int, db: Session, expires_at=None) -> GroupInviteOut: try:
token = secrets.token_urlsafe(16) invite = GroupInvite( group_id=group_id, created_by_id=user_id, token=token,
expires_at=expires_at ) db.add(invite) db.flush() db.refresh(invite) return invite
```

```
except Exception as e:
    db.rollback()
    logger.error(f"Error in group invite service: {e}")
    raise GroupInviteLinkCreateError from e
```

Frontend

```
===== FILE: frontend/.gitignore ===== #Logs logs .log npm-debug.log yarn-debug.log* yarn-error.log* pnpm-
debug.log* lerna-debug.log*
```

```
node_modules dist dist-ssr *.local
```

```
#Editor directories and files .vscode/* !.vscode/extensions.json .idea .DS_Store *.suo .ntvs *.njsproj *.sln *.sw?
```

```
===== FILE: frontend/Dockerfile ===== FROM node:23 AS build
```

```
WORKDIR /app
```

```
#Copy package files first COPY package*.json ./
```

```
#Install dependencies (legacy-peer-deps makes it more tolerant) RUN npm install --legacy-peer-deps
```

```
#Copy source code COPY . .
```

```
#Build Vite app RUN npm run build
```

```
#Production server: nginx FROM nginx:alpine COPY --from=build /app/dist /usr/share/nginx/html
```

```
EXPOSE 80 CMD ["nginx", "-g", "daemon off;"]
```

```
===== FILE: frontend/eslint.config.js ===== import js from '@eslint/js' import globals from 'globals' import
reactHooks from 'eslint-plugin-react-hooks' import reactRefresh from 'eslint-plugin-react-refresh' import {
defineConfig, globalIgnores } from 'eslint/config'
```

```
export default defineConfig([ globalIgnores(['dist']), { files: ['**/*.js,jsx'], extends: [ js.configs.recommended,
reactHooks.configs['recommended-latest'], reactRefresh.configs.vite, ], languageOptions: { ecmaVersion: 2020,
globals: globals.browser, parserOptions: { ecmaVersion: 'latest', ecmaFeatures: { jsx: true }, sourceType: 'module',
}, }, rules: { 'no-unused-vars': ['error', { varsIgnorePattern: '^[_A-Z_]' }], }, }, ])
```

```
===== FILE: frontend/index.html ===== <!doctype html>
```

```
===== FILE: frontend/package.json ===== { "name": "frontend", "private": true, "version": "0.0.0", "type":
"module", "scripts": { "dev": "vite", "build": "vite build", "lint": "eslint .", "preview": "vite preview" },
"dependencies": { "@radix-ui/react-checkbox": "^1.1.0", "@radix-ui/react-dialog": "^1.1.1", "@radix-ui/react-
select": "^2.1.2", "@radix-ui/react-slot": "^1.1.0", "@radix-ui/react-tabs": "^1.1.1", "class-variance-authority":
"^0.7.0", "clsx": "^2.1.1", "lucide-react": "^0.474.0", "react": "^19.1.1", "react-dom": "^19.1.1", "react-router-dom":
"^6.30.1", "tailwind-merge": "^2.5.2", "@radix-ui/react-label": "^2.1.0" }, "devDependencies": { "@eslint/js":
"^9.35.0", "@types/react": "^19.1.13", "@types/react-dom": "^19.1.9", "@vitejs/plugin-react": "^5.0.2",
"autoprefixer": "^10.4.20", "eslint": "^9.35.0", "eslint-plugin-react-hooks": "^5.2.0", "eslint-plugin-react-refresh":
```

```
"^0.4.20", "globals": "^16.4.0", "postcss": "^8.4.49", "tailwindcss": "^3.4.17", "tailwindcss-animate": "^1.0.7",
"vite": "^7.1.6" } }
```

```
===== FILE: frontend/package-lock.json ===== { "name": "frontend", "version": "0.0.0", "lockfileVersion": 3,
"requires": true, "packages": { "": { "name": "frontend", "version": "0.0.0", "dependencies": { "@radix-ui/react-
checkbox": "^1.1.0", "@radix-ui/react-dialog": "^1.1.1", "@radix-ui/react-label": "^2.1.0", "@radix-ui/react-select":
^2.1.2", "@radix-ui/react-slot": "^1.1.0", "@radix-ui/react-tabs": "^1.1.1", "class-variance-authority": "^0.7.0",
"clsx": "^2.1.1", "lucide-react": "^0.474.0", "react": "^19.1.1", "react-dom": "^19.1.1", "react-router-dom":
^6.30.1", "tailwind-merge": "^2.5.2", "devDependencies": { "@eslint/js": "^9.35.0", "@types/react": "^19.1.13",
"@types/react-dom": "^19.1.9", "@vitejs/plugin-react": "^5.0.2", "autoprefixer": "^10.4.20", "eslint": "^9.35.0",
"eslint-plugin-react-hooks": "^5.2.0", "eslint-plugin-react-refresh": "^0.4.20", "globals": "^16.4.0", "postcss":
^8.4.49", "tailwindcss": "^3.4.17", "tailwindcss-animate": "^1.0.7", "vite": "^7.1.6" } },
"node_modules/@alloc/quick-lru": { "version": "5.2.0", "resolved": "https://registry.npmjs.org/@alloc/quick-
lru/-/quick-lru-5.2.0.tgz", "integrity": "sha512-UrcABB+4bUrFABwbluTIBErXwvbsU/V7TZWfmbgJfbkwiBuziS9gxdODUyuiecdGQ85jglMW6juS3+z5TsKLw==",
"dev": true, "license": "MIT", "engines": { "node": ">=10" }, "funding": { "url":
"https://github.com/sponsors/sindresorhus" } }, "node_modules/@babel/code-frame": { "version": "7.27.1",
"resolved": "https://registry.npmjs.org/@babel/code-frame/-/code-frame-7.27.1.tgz", "integrity": "sha512-cjQ7ZlQ0Mv3b47hABuTevyTuYN4i+loJkGeV9flcCglK37cCXRh+L1bd3iBHlynerhQ7BhCkn2BPbQUL+rGqFg==",
"dev": true, "license": "MIT", "dependencies": { "@babel/helper-validator-identifier": "^7.27.1", "js-tokens":
^4.0.0", "picocolors": "^1.1.1" }, "engines": { "node": ">=6.9.0" } }, "node_modules/@babel/compat-data": {
"version": "7.28.4", "resolved": "https://registry.npmjs.org/@babel/compat-data/-/compat-data-7.28.4.tgz",
"integrity": "sha512-YsmSKC29MJwf0gF8Rjjrg5LQCmyh+j/nD8/eP7f+BeoQTKYqs9RoWbjGOdy0+1Ekr68RJZMUOPVQaQisnlo4Rw==",
"dev": true, "license": "MIT", "engines": { "node": ">=6.9.0" } }, "node_modules/@babel/core": { "version":
"7.28.4", "resolved": "https://registry.npmjs.org/@babel/core/-/core-7.28.4.tgz", "integrity": "sha512-2BCOP7TN8M+gVDj7/ht3hsaO/B/n5oDbiAyyvnRINOs+u1o+JWNYTQrmpuNp1/Wq2gcFrI01JAW+paEKDMx/CA==",
"dev": true, "license": "MIT", "dependencies": { "@babel/code-frame": "^7.27.1", "@babel/generator": "^7.28.3",
"@babel/helper-compilation-targets": "^7.27.2", "@babel/helper-module-transforms": "^7.28.3",
"@babel/helpers": "^7.28.4", "@babel/parser": "^7.28.4", "@babel/template": "^7.27.2", "@babel/traverse":
^7.28.4", "@babel/types": "^7.28.4", "@jridgewell/remapping": "^2.3.5", "convert-source-map": "^2.0.0",
"debug": "^4.1.0", "gensync": "^1.0.0-beta.2", "json5": "^2.2.3", "semver": "^6.3.1" }, "engines": { "node":
">=6.9.0" }, "funding": { "type": "opencollective", "url": "https://opencollective.com/babel" } },
"node_modules/@babel/generator": { "version": "7.28.3", "resolved":
"https://registry.npmjs.org/@babel/generator/-/generator-7.28.3.tgz", "integrity": "sha512-3ISpxGgvnmZznmBkCRnVREPUFJv2wrv9iAoFDvADJc0ypmdOxdUtlcBgBJ6zE0PMeTKnxQzyk0xBq4Ep7zw==",
"dev": true, "license": "MIT", "dependencies": { "@babel/parser": "^7.28.3", "@babel/types": "^7.28.2",
"@jridgewell/gen-mapping": "^0.3.12", "@jridgewell/trace-mapping": "^0.3.28", "jsesc": "^3.0.2" }, "engines": {
"node": ">=6.9.0" } }, "node_modules/@babel/helper-compilation-targets": { "version": "7.27.2", "resolved":
"https://registry.npmjs.org/@babel/helper-compilation-targets/-/helper-compilation-targets-7.27.2.tgz", "integrity":
sha512-sha512-
2+1thGUUWWjLTYTHZWk1n8Yga0ijBz1XAhUXcKy81rd5g6yh7hGqMp45v7cadSbEHc9G3OTv45SyneRN3ps4DQ=
=", "dev": true, "license": "MIT", "dependencies": { "@babel/compat-data": "^7.27.2", "@babel/helper-validator-
option": "^7.27.1", "browserslist": "^4.24.0", "lru-cache": "^5.1.1", "semver": "^6.3.1" }, "engines": { "node":
">=6.9.0" } }, "node_modules/@babel/helper-globals": { "version": "7.28.0", "resolved":
"https://registry.npmjs.org/@babel/helper-globals/-/helper-globals-7.28.0.tgz", "integrity": "sha512-
+W6clSkXFa1jXsDEdYA8HeevQT/FULhxzR99pxphltZcVaugps53THCeiWA8SguxxpSp3gKPiuYfSWopkLQ4hw==",
"dev": true, "license": "MIT", "engines": { "node": ">=6.9.0" } }, "node_modules/@babel/helper-module-imports": {
"version": "7.27.1", "resolved": "https://registry.npmjs.org/@babel/helper-module-imports/-/helper-module-
```

imports-7.27.1.tgz", "integrity": "sha512-0gSFWUPNXNopqtlIPQvID5WgXYI5GY2kP2cCvoT8kczjbfcfulljTbcWrulD1CIPX2gt1wghbDy08yE1p+/r3w==", "dev": true, "license": "MIT", "dependencies": { "@babel/traverse": "^7.27.1", "@babel/types": "^7.27.1" }, "engines": { "node": ">=6.9.0" }, "node_modules/@babel/helper-module-transforms": { "version": "7.28.3", "resolved": "https://registry.npmjs.org/@babel/helper-module-transforms/-/helper-module-transforms-7.28.3.tgz", "integrity": "sha512-gytXUbs8k2sXS9PnQptz5o0QnpLL51SwASIORY6XaBKF88nsOT0Zw9szLqISGQDP/4TljBAD5y98p2U1fqkds==", "dev": true, "license": "MIT", "dependencies": { "@babel/helper-module-imports": "^7.27.1", "@babel/helper-validator-identifier": "^7.27.1", "@babel/traverse": "^7.28.3" }, "engines": { "node": ">=6.9.0" }, "peerDependencies": { "@babel/core": "^7.0.0" }, "node_modules/@babel/helper-plugin-utils": { "version": "7.27.1", "resolved": "https://registry.npmjs.org/@babel/helper-plugin-utils/-/helper-plugin-utils-7.27.1.tgz", "integrity": "sha512-1gn1Up5YXka3YYAHGKpbideQ5Yjf1tDa9qYcgysz+cNCXukyLl6DjPXhD3VRwSb8c0J9tA4b2+rHEZtc6R0tlw==", "dev": true, "license": "MIT", "engines": { "node": ">=6.9.0" }, "node_modules/@babel/helper-string-parser": { "version": "7.27.1", "resolved": "https://registry.npmjs.org/@babel/helper-string-parser/-/helper-string-parser-7.27.1.tgz", "integrity": "sha512-qMSxKbpRIArIdDExk92nSobyDdpPijUq2DW6oDnUqd0iOGxmQjyqhMliH9+zv4LPyZdRje2cavWPbCbWm3eA==", "dev": true, "license": "MIT", "engines": { "node": ">=6.9.0" }, "node_modules/@babel/helper-validator-identifier": { "version": "7.27.1", "resolved": "https://registry.npmjs.org/@babel/helper-validator-identifier/-/helper-validator-identifier-7.27.1.tgz", "integrity": "sha512-D2hP9eA+Sqx1kBZgzxZh0y1trbuU+JoDkiEwqhQ36nodYqJwyElhPSdMNd7IOm/4io72luTPWH20Yda0xOuUow==", "dev": true, "license": "MIT", "engines": { "node": ">=6.9.0" }, "node_modules/@babel/helper-validator-option": { "version": "7.27.1", "resolved": "https://registry.npmjs.org/@babel/helper-validator-option/-/helper-validator-option-7.27.1.tgz", "integrity": "sha512-YvjJow9FxbhFFKDSuFnVcE2WxXk1zWc22fFePVNEaWJEU8lrZVIda6N0uHwzZrUM1il7NC9Mlp4MaJYbYd9JSg==", "dev": true, "license": "MIT", "engines": { "node": ">=6.9.0" }, "node_modules/@babel/helpers": { "version": "7.28.4", "resolved": "https://registry.npmjs.org/@babel/helpers/-/helpers-7.28.4.tgz", "integrity": "sha512-HFN59MmQXGHVyYadKLVumYsA9dBFun/lDYxipEjzA4196jpLZd8UJEEBLkbEkvfYreDqJhZxYAWFPtrfhNpj4w==", "dev": true, "license": "MIT", "dependencies": { "@babel/template": "^7.27.2", "@babel/types": "^7.28.4" }, "engines": { "node": ">=6.9.0" }, "node_modules/@babel/parser": { "version": "7.28.4", "resolved": "https://registry.npmjs.org/@babel/parser/-/parser-7.28.4.tgz", "integrity": "sha512-yZbBqeM6TkP9du/l2pUZnJsRMGGvOulrhjzC1AwHwW+6he4mni6Bp/m8ijn0iOuZuPI2BfkCoSRunpyjnrQKg==", "dev": true, "license": "MIT", "dependencies": { "@babel/types": "^7.28.4" }, "bin": { "parser": "bin/babel-parser.js" }, "engines": { "node": ">=6.0.0" }, "node_modules/@babel/plugin-transform-react-jsx-self": { "version": "7.27.1", "resolved": "https://registry.npmjs.org/@babel/plugin-transform-react-jsx-self/-/plugin-transform-react-jsx-self-7.27.1.tgz", "integrity": "sha512-6UzkCs+ejGdZ5mFFC/OCUrv028ab2fp1znZmCZjAOBKibK2jXD1O+BPSfX8X2qjJ75fZBMSnQn3Rq2mrBJK2mw==", "dev": true, "license": "MIT", "dependencies": { "@babel/helper-plugin-utils": "^7.27.1", "engines": { "node": ">=6.9.0" }, "peerDependencies": { "@babel/core": "^7.0.0-0" }, "node_modules/@babel/plugin-transform-react-jsx-source": { "version": "7.27.1", "resolved": "https://registry.npmjs.org/@babel/plugin-transform-react-jsx-source/-/plugin-transform-react-jsx-source-7.27.1.tgz", "integrity": "sha512-zbwoTsBruTeKB9hSq73ha66iFeJHuaFkUbwvqElnygoNbjjHRsSeokowZFN3CZ64lvEqcmmkVe89OPXc7ldAw==", "dev": true, "license": "MIT", "dependencies": { "@babel/helper-plugin-utils": "^7.27.1", "engines": { "node": ">=6.9.0" }, "peerDependencies": { "@babel/core": "^7.0.0-0" }, "node_modules/@babel/template": { "version": "7.27.2", "resolved": "https://registry.npmjs.org/@babel/template/-/template-7.27.2.tgz", "integrity": "sha512-LPDZ85aEjYYSd18/DkjNh4/y1ntkE5KwUHWtiqqRxruuZL2F1yuHligVHLvcHY2vMHXttKFpJn6Lwfl7cw7ODw==", "dev": true, "license": "MIT", "dependencies": { "@babel/code-frame": "^7.27.1", "@babel/parser": "^7.27.2", "@babel/types": "^7.27.1", "engines": { "node": ">=6.9.0" }, "node_modules/@babel/traverse": { "version": "7.28.4", "resolved": "https://registry.npmjs.org/@babel/traverse/-/traverse-7.28.4.tgz", "integrity": "sha512-5611z05HbV8YSp492TgyHHuTDs8h7az13a22VNf/3s8R1UpTY7t1Y5o1ahLr2zL7tCl/BgjZ141loAQA=="

```
YEzuboP2qvQavAcjgQNVgsvHIDv6ZpwXvcvjmyySP2DIMuByS/6ioU5G9pYrWHM6T2YDfc7xga9iNzYOs12CFQ==",
"dev": true, "license": "MIT", "dependencies": { "@babel/code-frame": "^7.27.1", "@babel/generator": "^7.28.3",
"@babel/helper-globals": "^7.28.0", "@babel/parser": "^7.28.4", "@babel/template": "^7.27.2", "@babel/types":
"^7.28.4", "debug": "^4.3.1" }, "engines": { "node": ">=6.9.0" } }, "node_modules/@babel/types": { "version":
"7.28.4", "resolved": "https://registry.npmjs.org/@babel/types/-/types-7.28.4.tgz", "integrity": "sha512-
bkFqkLhh3pMBUQQkpVgWDWq/lqzc2678eUyDITBhRqhCHFGuYYGM0Efga7tYk4TogG/3x0EEI66/OQ+WGbWB/Q=
=", "dev": true, "license": "MIT", "dependencies": { "@babel/helper-string-parser": "^7.27.1", "@babel/helper-
validator-identifier": "^7.27.1" }, "engines": { "node": ">=6.9.0" } }, "node_modules/@esbuild/aix-ppc64": {
"version": "0.25.10", "resolved": "https://registry.npmjs.org/@esbuild/aix-ppc64/-/aix-ppc64-0.25.10.tgz",
"integrity": "sha512-
0NFWnA+7141irNuaSVILfgNT12caWJVLzp5eAVhZ0z1qpxbockccEt3s+149rE64VUI3MI2zt8Nv5JVc4QXTsw==",
"cpu": [ "ppc64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "aix" ], "engines": { "node": ">=18" } },
"node_modules/@esbuild/android-arm": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/android-arm/-/android-arm-0.25.10.tgz", "integrity": "sha512-
dQAx1dW1C3zpeCDc5KqIYuZ1tgAdRXNoZP7vkBIRtKZPYe2xVr/d3SkirkICHudW1B45tGiUlz2pUWdfbDD4w==",
"cpu": [ "arm" ], "dev": true, "license": "MIT", "optional": true, "os": [ "android" ], "engines": { "node": ">=18" } },
"node_modules/@esbuild/android-arm64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/android-arm64/-/android-arm64-0.25.10.tgz", "integrity": "sha512-
LSQa7eDahypv/VO6WKohZGPSJDq5OVOo3UoFR1E4t4Gj1W7zEQMUhl+lo81H+DtB+kP+tDgBp+M4oNCwp6kffg=
=", "cpu": [ "arm64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "android" ], "engines": { "node":
">=18" } }, "node_modules/@esbuild/android-x64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/android-x64/-/android-x64-0.25.10.tgz", "integrity": "sha512-
MiC9CWdPrfhibcXwr39p9ha1x0IZJ9KaVfvzA0Wxwz9ETX4v5CHfF09bx935nHlhi+MxhA63dKRRQLiVgSUtEg==",
"cpu": [ "x64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "android" ], "engines": { "node": ">=18" } },
"node_modules/@esbuild/darwin-arm64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/darwin-arm64/-/darwin-arm64-0.25.10.tgz", "integrity": "sha512-
JC74bdXcQEpW9KkV326WpZZJLguSZ3DfS8wrrvPMHgQOIEIG/sPXEN/V8IssoJhbefLRcRqw6RQH2NnpdprtMA==
", "cpu": [ "arm64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "darwin" ], "engines": { "node": ">=18"
} }, "node_modules/@esbuild/darwin-x64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/darwin-x64/-/darwin-x64-0.25.10.tgz", "integrity": "sha512-
tguWg1oIF6DGqzws97pKZ8G2L7lg1vjDmGTwcTuYHbuU6TTJJe5FXbgs5C1BBzHbJ2bo1m3WkQDbWO2PvamRcg
==", "cpu": [ "x64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "darwin" ], "engines": { "node": ">=18"
} }, "node_modules/@esbuild/freebsd-arm64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/freebsd-arm64/-/freebsd-arm64-0.25.10.tgz", "integrity": "sha512-
3ZioSQSg1HT2N05YxeJWYR+Libe3bREVSDWhEEGExWaDtyFbbXWb49QgPvFH8u03vUPX10JhJPcz7s9t9+boWg
==", "cpu": [ "arm64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "freebsd" ], "engines": { "node":
">=18" } }, "node_modules/@esbuild/freebsd-x64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/freebsd-x64/-/freebsd-x64-0.25.10.tgz", "integrity": "sha512-
LLgJfHJk014Aa4anGDbh8bml5Lk+QidDmGzuC2D+vP7mv/GeSN+H39zOf7pN5N8p059FcOfs2bVlrRr4SK9WxA==
", "cpu": [ "x64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "freebsd" ], "engines": { "node": ">=18"
} }, "node_modules/@esbuild/linux-arm": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/linux-arm/-/linux-arm-0.25.10.tgz", "integrity": "sha512-
oR31GtBTfYcQEBALi9r6WxoU/ZofZl962pouZRTEYECvNF/dtXKku8YXcJkhgK/beU+zedXflzHijSRapJY3vg==",
"cpu": [ "arm" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ], "engines": { "node": ">=18" } },
"node_modules/@esbuild/linux-arm64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/linux-arm64/-/linux-arm64-0.25.10.tgz", "integrity": "sha512-
5luJWN6YKbsawd5f9i4+c+geYiVEw20FVW5x0v1kEMWNq8UctFjDiMATBxLvmmHA4bf7F6hTRaJgtghFr9iziQ==",
"cpu": [ "arm64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ], "engines": { "node": ">=18" } },
"node_modules/@esbuild/linux-ia32": { "version": "0.25.10", "resolved": "https://registry.npmjs.org/@esbuild/linux-

```

```
ia32/-linux-ia32-0.25.10.tgz", "integrity": "sha512-
NrSCx2Kim3EnnWgS4Txn0QGt0Xipoumb6z6sUtl5bOEZIVKhzfyp/Lyw4C1DIYvzeW/5mWYPBFJU3a/8Yr75DQ==",
"cpu": [ "ia32" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ], "engines": { "node": ">=18" } },
"node_modules/@esbuild/linux-loong64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/linux-loong64/-linux-loong64-0.25.10.tgz", "integrity": "sha512-
xoSphrd4AZda8+rUDDfD9J6FUMjrkTz8itpTITM4/xgerAZZcFW7Dv+sun7333lfKxGG8gAq+3NbfEMJfiY+Eg==",
"cpu": [ "loong64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ], "engines": { "node": ">=18" } },
"node_modules/@esbuild/linux-mips64el": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/linux-mips64el/-linux-mips64el-0.25.10.tgz", "integrity": "sha512-
ab6eiuCwoMmYDyTnyptoKkVS3k8fy/1Uvq7Dj5czXI6DF2GqD2ToInBI0SHOp5/X1BdZ26RKc5+qjQNGRBeIRA==",
"cpu": [ "mips64el" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ], "engines": { "node": ">=18" }
}, "node_modules/@esbuild/linux-ppc64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/linux-ppc64/-linux-ppc64-0.25.10.tgz", "integrity": "sha512-
NLinzzOgZQsGpsTkEbdJTCanwA5/wozN9dSgEI12haXJBzMTpssebuXR42bthOF3z7zXFWH1AmvWunUCkBE4EA=
=", "cpu": [ "ppc64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ], "engines": { "node": ">=18" }
}, "node_modules/@esbuild/linux-riscv64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/linux-riscv64/-linux-riscv64-0.25.10.tgz", "integrity": "sha512-
FE557XdZDrtX8NMleA8LBJX3dC2M8VGXwfrQWU7LB5SLOajfJlXmSdyL/gU1m64Zs9CBKvm4UAuBp5aJ8OgnrA=
=", "cpu": [ "riscv64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ], "engines": { "node": ">=18"
} }, "node_modules/@esbuild/linux-s390x": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/linux-s390x/-linux-s390x-0.25.10.tgz", "integrity": "sha512-
3BBSbgzuB9ajLoVZk0mGu+EHIBwkusRmeNYdqmznmMc9zGASFjSsxgkNsqmXugpPk00gJ0JNKh/97nxmjctdew=
=", "cpu": [ "s390x" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ], "engines": { "node": ">=18" }
}, "node_modules/@esbuild/linux-x64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/linux-x64/-linux-x64-0.25.10.tgz", "integrity": "sha512-
QsX81KhFoZGwenVyPoberggdW1nrQZSvfVDAIUxR3WqLRZGZqWk/P4T8p2SP+de2Sr5HPcvjhcJzEiulKgnxtA==",
"cpu": [ "x64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ], "engines": { "node": ">=18" } },
"node_modules/@esbuild/netbsd-arm64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/netbsd-arm64/-netbsd-arm64-0.25.10.tgz", "integrity": "sha512-
AKQM3gfYfSW8XRk8DdMCzaLUFB15dTrZfnX8WXQoOUpUBQ+NaAFCP1kPS/ykbbGYz7rxn0WS48/81I9hFI3u4A=
=", "cpu": [ "arm64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "netbsd" ], "engines": { "node":
">=18" } }, "node_modules/@esbuild/netbsd-x64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/netbsd-x64/-netbsd-x64-0.25.10.tgz", "integrity": "sha512-
7RTytDPGU6fek/hWuN9qQpeGPBZFfB4zZgcZ2VK2Z5VpdUxEI8JKYsg3JfO0n/Z1E/6l05n0unDCNc4HnhQGig==",
"cpu": [ "x64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "netbsd" ], "engines": { "node": ">=18" } },
"node_modules/@esbuild/openbsd-arm64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/openbsd-arm64/-openbsd-arm64-0.25.10.tgz", "integrity": "sha512-
5Se0VM9Wtq797YFn+dLimf2Zx6McttsH2oLUBsDml+lm0GOCRvEbRWUvDtKY4BWYv/3NgzS8b/UM3jQNh5hYyw=
=", "cpu": [ "arm64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "openbsd" ], "engines": { "node":
">=18" } }, "node_modules/@esbuild/openbsd-x64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/openbsd-x64/-openbsd-x64-0.25.10.tgz", "integrity": "sha512-
XkA4frq1TLj4bEMB+2HnI0+4RnjbuGZfet2gs/LNs5Hc7D89ZQBHQ0gL2ND6Lzu1+QVkj3x1gIcPKzRNP8bXw==",
"cpu": [ "x64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "openbsd" ], "engines": { "node": ">=18" } },
"node_modules/@esbuild/openharmony-arm64": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/@esbuild/openharmony-arm64/-openharmony-arm64-0.25.10.tgz", "integrity":
"sha512-
AVTSBhTX8Y/Fz6OmIVBip9tJzZEUcY8WLh7I59+upa5/GPhh2/aM6bvOMQySspnCCHvFi79kMtdJS1w0DXAeag==",
"cpu": [ "arm64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "openharmony" ], "engines": { "node":
">=18" } }, "node_modules/@esbuild/sunos-x64": { "version": "0.25.10", "resolved":
```

"https://registry.npmjs.org/@esbuild/sunos-x64/-/sunos-x64-0.25.10.tgz", "integrity": "sha512-fswk3XT0Uf2pGJmOpDB7yknqhVkJQkAQOcW/ccVOtfx05LkbWOaRAtn5SaqXypeKQra1QaEa841PgrSL9ubSPQ==", "cpu": ["x64"], "dev": true, "license": "MIT", "optional": true, "os": ["sunos"], "engines": { "node": ">=18" } }, "node_modules/@esbuild/win32-arm64": { "version": "0.25.10", "resolved": "https://registry.npmjs.org/@esbuild/win32-arm64/-/win32-arm64-0.25.10.tgz", "integrity": "sha512-ah+9b59KDTsfpaCg6VdJoOQvKjI33nTaQr4UluQwW7aEwZQsbMcFTmfEO4VyewOxx4RaDT/xCy9ra2GPWmO7Kw==", "cpu": ["arm64"], "dev": true, "license": "MIT", "optional": true, "os": ["win32"], "engines": { "node": ">=18" } }, "node_modules/@esbuild/win32-ia32": { "version": "0.25.10", "resolved": "https://registry.npmjs.org/@esbuild/win32-ia32/-/win32-ia32-0.25.10.tgz", "integrity": "sha512-QHPDbKkrGO8/cz9LKVnJU22HOi4pxZnZhhA2HYHez5Pz4JeffhDjf85E57Oyco163GnzNCVkJZK0b/n4Y0UHcSw==", "cpu": ["ia32"], "dev": true, "license": "MIT", "optional": true, "os": ["win32"], "engines": { "node": ">=18" } }, "node_modules/@esbuild/win32-x64": { "version": "0.25.10", "resolved": "https://registry.npmjs.org/@esbuild/win32-x64/-/win32-x64-0.25.10.tgz", "integrity": "sha512-9KpxSVFCu0iK1owoez6aC/s/EdUQLDN3adTxGCqxMVhrPDj6bt5dbrHDXUuq+Bs2vATFBBRQS5vdQ/Ed2P+nbw==", "cpu": ["x64"], "dev": true, "license": "MIT", "optional": true, "os": ["win32"], "engines": { "node": ">=18" } }, "node_modules/@eslint-community/eslint-utils": { "version": "4.9.0", "resolved": "https://registry.npmjs.org/@eslint-community/eslint-utils/-/eslint-utils-4.9.0.tgz", "integrity": "sha512-ayVFHdtZ+hsq1t2Dy24wCmGXGe4q9Gu3smhLYALJrr473ZH27MsnSL+LKUlimp4BWJqMDMLmPpx/Q9R3OAIL4g==", "dev": true, "license": "MIT", "dependencies": { "eslint-visitor-keys": "^3.4.3" }, "engines": { "node": ">=12.22.0 || ^14.17.0 || >=16.0.0" }, "funding": { "url": "https://opencollective.com/eslint" }, "peerDependencies": { "eslint": ">=6.0.0 || ^7.0.0 || >=8.0.0" }, "node_modules/@eslint-community/eslint-utils/node_modules/eslint-visitor-keys": { "version": "3.4.3", "resolved": "https://registry.npmjs.org/eslint-visitor-keys/-/eslint-visitor-keys-3.4.3.tgz", "integrity": "sha512-wpc+LXeiYisxPIEkUzU6svyS1frio3Mgxj1fdy7Pm8Ygzguax2N3Fa/D/ag1WqbOprdi+uY6wMUI8/a2G+iag==", "dev": true, "license": "Apache-2.0", "engines": { "node": ">=12.22.0 || ^14.17.0 || >=16.0.0" }, "funding": { "url": "https://opencollective.com/eslint" } }, "node_modules/@eslint-community/regexpp": { "version": "4.12.1", "resolved": "https://registry.npmjs.org/@eslint-community/regexpp/-/regexpp-4.12.1.tgz", "integrity": "sha512-CCZCDJuduB9OUkFkY2lgppNZMi2IBQgD2qzwXkEia16cge2pijY/aXi96CJMquDMn3nJdlPV1A5KrJEXwflNzQ==", "dev": true, "license": "MIT", "engines": { "node": ">=12.0.0 || ^14.0.0 || >=16.0.0" } }, "node_modules/@eslint/config-array": { "version": "0.21.0", "resolved": "https://registry.npmjs.org/@eslint/config-array/-/config-array-0.21.0.tgz", "integrity": "sha512-ENIdciLu0d93HeYirvKmrzshzofPw6VkJZKQGe9Nv46ZnWUzcF1xV01dcvEg/1wXUR61OmmlSfyeyO7EvjLxQ==", "dev": true, "license": "Apache-2.0", "dependencies": { "@eslint/object-schema": "^2.1.6", "debug": "^4.3.1", "minimatch": "^3.1.2" }, "engines": { "node": ">=18.18.0 || ^20.9.0 || >=21.1.0" }, "node_modules/@eslint/config-helpers": { "version": "0.3.1", "resolved": "https://registry.npmjs.org/@eslint/config-helpers/-/config-helpers-0.3.1.tgz", "integrity": "sha512-xR93k9WhrDYpXHORXpxVL5oHj3Era7wo6k/Wd8/IsQNNZUTzkGS29lyn3nAT05v6ltUuTFVCCYDEGfy2Or/sPA==", "dev": true, "license": "Apache-2.0", "engines": { "node": ">=18.18.0 || ^20.9.0 || >=21.1.0" } }, "node_modules/@eslint/core": { "version": "0.15.2", "resolved": "https://registry.npmjs.org/@eslint/core/-/core-0.15.2.tgz", "integrity": "sha512-78Md3/Rrxh83gCxoUc0EiciuOHsITzLy53m3d9UyiW8y9Dj2D29FeETqyKA+BRK76tnTp6RXWb3pCay8Oyomg==", "dev": true, "license": "Apache-2.0", "dependencies": { "@types/json-schema": "^7.0.15" }, "engines": { "node": ">=18.18.0 || ^20.9.0 || >=21.1.0" }, "node_modules/@eslint/eslintrc": { "version": "3.3.1", "resolved": "https://registry.npmjs.org/@eslint/eslintrc/-/eslintrc-3.3.1.tgz", "integrity": "sha512-gtF186CXhl1p4pJNGZw8Yc6RlshoePRvE0X91oPGb3vZ8pM3qOS9W9NGPat9LziaBV7XrJWGyINQXkGcnM3IQ==", "dev": true, "license": "MIT", "dependencies": { "ajv": "^6.12.4", "debug": "^4.3.2", "esprez": "^10.0.1", "globals": "^14.0.0", "ignore": "^5.2.0", "import-fresh": "^3.2.1", "js-yaml": "^4.1.0", "minimatch": "^3.1.2", "strip-json-comments": "^3.1.1" }, "engines": { "node": ">=18.18.0 || ^20.9.0 || >=21.1.0" }, "funding": { "url": "https://opencollective.com/eslint" } }, "node_modules/@eslint/eslintrc/node_modules/globals": { "version": "14.0.0", "resolved": "https://registry.npmjs.org/globals/-/globals-14.0.0.tgz", "integrity": "sha512-1v8G0JkAebKiv4U4Lgtg91YwbhRQm082CsWAbN7BKlnKwpxW4YUFBHG34WoN19Wgmw9TUl4Yqz5/Z540LQ=="

"14.0.0", "resolved": "https://registry.npmjs.org/globals/-/globals-14.0.0.tgz", "integrity": "sha512-oahGvuMGQIPw/ivIYBjVSrWAFWLBeku5tpPE2fOPLi+WHfflWbuh2tCjhyQhTBPMf5E9jDEH4FOMTYgYwbKwtQ==", "dev": true, "license": "MIT", "engines": { "node": ">=18" }, "funding": { "url": "https://github.com/sponsors/sindresorhus" } }, "node_modules/@eslint/js": { "version": "9.35.0", "resolved": "https://registry.npmjs.org/@eslint/js/-/js-9.35.0.tgz", "integrity": "sha512-30iXE9whjllFwobBkNerJo+TXYsgVM5ERQwMcMKCHckHfCmf7wXDAHIARoWnh0s1U72WqlbeyE7iAcCzuCPw==", "dev": true, "license": "MIT", "engines": { "node": "^18.18.0 || ^20.9.0 || >=21.1.0" }, "funding": { "url": "https://eslint.org/donate" } }, "node_modules/@eslint/object-schema": { "version": "2.1.6", "resolved": "https://registry.npmjs.org/@eslint/object-schema/-/object-schema-2.1.6.tgz", "integrity": "sha512-RBMg5FRL0I0gs51M/guSAj5/e14VQ4tpZnQNWwuDT66P14I43ltmPfIZRhO9fUVIPOAQXU47atlywZ/czoqFPA==", "dev": true, "license": "Apache-2.0", "engines": { "node": "^18.18.0 || ^20.9.0 || >=21.1.0" } }, "node_modules/@eslint/plugin-kit": { "version": "0.3.5", "resolved": "https://registry.npmjs.org/@eslint/plugin-kit/-/plugin-kit-0.3.5.tgz", "integrity": "sha512-Z5kJ+wU3oA7MMlqVR9tyZRtjYPr4OC004Q4Rw7pgOKUOKkJfZ3O24nz3WYfGRpMDNmcOi3TwQOmgm7B7Tpii0w==", "dev": true, "license": "Apache-2.0", "dependencies": { "@eslint/core": "^0.15.2", "levn": "^0.4.1" }, "engines": { "node": "^18.18.0 || ^20.9.0 || >=21.1.0" } }, "node_modules/@floating-ui/core": { "version": "1.7.3", "resolved": "https://registry.npmjs.org/@floating-ui/core/-/core-1.7.3.tgz", "integrity": "sha512-sGnvb5dmrJaKEZ+LDlpguvdx3bDIEllmv4/CiQ9awcmCZrlx5jQyyMWFM5kBI+EyNOCDDiKk8il0zeuX3Zlg/w==", "license": "MIT", "dependencies": { "@floating-ui/utils": "^0.2.10" } }, "node_modules/@floating-ui/dom": { "version": "1.7.4", "resolved": "https://registry.npmjs.org/@floating-ui/dom/-/dom-1.7.4.tgz", "integrity": "sha512-OOchDgh4F2CchOX94cRVqhyv7b3AFb+/rQXyswmzmGakRfkMgoWVjfnLWkRirfLEfuD4ysVW16eXzwt3jHlzKA==", "license": "MIT", "dependencies": { "@floating-ui/core": "^1.7.3", "@floating-ui/utils": "^0.2.10" } }, "node_modules/@floating-ui/react-dom": { "version": "2.1.6", "resolved": "https://registry.npmjs.org/@floating-ui/react-dom/-/react-dom-2.1.6.tgz", "integrity": "sha512-4JX6rEatQEvlmgU80wZyq9RT96HZJa88q8hp0pBd+LrczeDI4o6uA2M+uvxngVHo4lhr8uibXxH6+70zhAFrVw==", "license": "MIT", "dependencies": { "@floating-ui/dom": "^1.7.4", "peerDependencies": { "react": ">=16.8.0", "react-dom": ">=16.8.0" } }, "node_modules/@floating-ui/utils": { "version": "0.2.10", "resolved": "https://registry.npmjs.org/@floating-ui/utils/-/utils-0.2.10.tgz", "integrity": "sha512-aGTxbpb8/b5JfU1HXSrbH3wXZuLPJcNEcZQFMxL3oSzgtVu6nFPkbbGGUvBcUjKV2YyB9Wxxabo+HEH9tcRQ==", "license": "MIT" }, "node_modules/@humanfs/core": { "version": "0.19.1", "resolved": "https://registry.npmjs.org/@humanfs/core/-/core-0.19.1.tgz", "integrity": "sha512-5DyQ4+1JEUzejeK1JGICcideyfUbGixgS9jNgex5nqkW+cY7WZhxBigmieN5Qnw9ZosSNVC9KQKyb+GUaGyKUA==", "dev": true, "license": "Apache-2.0", "engines": { "node": ">=18.18.0" } }, "node_modules/@humanfs/node": { "version": "0.16.7", "resolved": "https://registry.npmjs.org/@humanfs/node/-/node-0.16.7.tgz", "integrity": "sha512-zUx+yOslrG4Y43Eh2peDeKCxIRt/gET6aHfaKpuq267qXdyDFViVHfMaLyygZOnI0kGWxFlgsBy8QFuTLUXEQ==", "dev": true, "license": "Apache-2.0", "dependencies": { "@humanfs/core": "^0.19.1", "@humanwhocodes/retry": "^0.4.0" }, "engines": { "node": ">=18.18.0" } }, "node_modules/@humanwhocodes/module-importer": { "version": "1.0.1", "resolved": "https://registry.npmjs.org/@humanwhocodes/module-importer/-/module-importer-1.0.1.tgz", "integrity": "sha512-bxveV4V8v5Yb4ncFTT3rPSgZBOpCkjjfK0y4oVVVJwluDVBRMDXrPyXRL988i5ap9m9bnyEEjWfm5WkBmtffLfA==", "dev": true, "license": "Apache-2.0", "engines": { "node": ">=12.22" }, "funding": { "type": "github", "url": "https://github.com/sponsors/nzakas" } }, "node_modules/@humanwhocodes/retry": { "version": "0.4.3", "resolved": "https://registry.npmjs.org/@humanwhocodes/retry/-/retry-0.4.3.tgz", "integrity": "sha512-bV0Tgo9K4hfPCek+aMA81RppFKv2ySDQeMoSZuvTASywnTnVJCArCZE2FWqpvlAtKu7VMRLWIR1EazvVhDyhQ==", "dev": true, "license": "Apache-2.0", "engines": { "node": ">=18.18" }, "funding": { "type": "github", "url": "https://github.com/sponsors/nzakas" } }, "node_modules/@isaacs/cliui": { "version": "8.0.2", "resolved": "https://registry.npmjs.org/@isaacs/cliui/-/cliui-8.0.2.tgz", "integrity": "sha512-O8jcjabXaleOG9DQ0+ARXWZBTfnP4WNAqzuiJK7II44AmxGKv/J2M4TPjxjY3znBCfvBXFzucm1twdyFybfqEA==", "dev": true, "license": "ISC", "dependencies": { "string-width": "^5.1.2", "string-width-cjs": "npm:string-"

```

width@^4.2.0", "strip-ansi": "^7.0.1", "strip-ansi-cjs": "npm:strip-ansi@^6.0.1", "wrap-ansi": "^8.1.0", "wrap-ansi-cjs": "npm:wrap-ansi@^7.0.0", "engines": { "node": ">=12" }, "node_modules/@jridgewell/gen-mapping": { "version": "0.3.13", "resolved": "https://registry.npmjs.org/@jridgewell/gen-mapping/-/gen-mapping-0.3.13.tgz", "integrity": "sha512-2kkt/7niJ6MgEPxF0bYdQ6etZaA+fQvDcLKckhy1yIQOzaoKjBBjSj63/aLVjYE3qhRt5dvM+uUyfCg6UKCBbA==", "dev": true, "license": "MIT", "dependencies": { "@jridgewell/sourcemap-codec": "^1.5.0", "@jridgewell/trace-mapping": "^0.3.24" }, "node_modules/@jridgewell/remapping": { "version": "2.3.5", "resolved": "https://registry.npmjs.org/@jridgewell/remapping/-/remapping-2.3.5.tgz", "integrity": "sha512-LI9u/+laYG4Ds1TDKSJW2YPrllcVYOwi2fUC6xB43lueCjgxV4lffOCZCtYFiH6TNOX+tQKXx97T4IKHbhyHEQ==", "dev": true, "license": "MIT", "dependencies": { "@jridgewell/gen-mapping": "^0.3.5", "@jridgewell/trace-mapping": "^0.3.24" }, "node_modules/@jridgewell/resolve-uri": { "version": "3.1.2", "resolved": "https://registry.npmjs.org/@jridgewell/resolve-uri/-/resolve-uri-3.1.2.tgz", "integrity": "sha512-bRISgCljP20/tbWSPWMEi54QVPRZExkuD9IJL+UlxUKtwVJA8wW1Trb1jMs1RFXo1CBTNZ/5hpC9QvmKWdopKw==", "dev": true, "license": "MIT", "engines": { "node": ">=6.0.0" }, "node_modules/@jridgewell/sourcemap-codec": { "version": "1.5.5", "resolved": "https://registry.npmjs.org/@jridgewell/sourcemap-codec/-/sourcemap-codec-1.5.5.tgz", "integrity": "sha512-cYQ9310grqxueWbl+WuIUaiUaDcj7WOq5fVhEljNVgRfOUhY9fy2zTvfoqWsnebh8SI70VScFbICvJnLKB0Og==", "dev": true, "license": "MIT", "node_modules/@jridgewell/trace-mapping": { "version": "0.3.31", "resolved": "https://registry.npmjs.org/@jridgewell/trace-mapping/-/trace-mapping-0.3.31.tgz", "integrity": "sha512-zzNR+SdQSDJzc8joaeP8QQoCQR8NuYx2dIlyt1QeBEZHJ9uW6hebsrYgbz8hJwUQao3TWCMtmfV8Nu1twOLAw==", "dev": true, "license": "MIT", "dependencies": { "@jridgewell/resolve-uri": "^3.1.0", "@jridgewell/sourcemap-codec": "^1.4.14" }, "node_modules/@nodelib/fs.scandir": { "version": "2.1.5", "resolved": "https://registry.npmjs.org/@nodelib/fs.scandir/-/fs.scandir-2.1.5.tgz", "integrity": "sha512-vq24Bq3ym5HEQm2NKCrc3yXDwc7vTsEThRDnkp2DK9p1uqLR+DHurm/NOTo0KG7HYHU7eppKZj3MyqYuMBf62g==", "dev": true, "license": "MIT", "dependencies": { "@nodelib/fs.stat": "2.0.5", "run-parallel": "^1.1.9" }, "engines": { "node": ">= 8" }, "node_modules/@nodelib/fs.stat": { "version": "2.0.5", "resolved": "https://registry.npmjs.org/@nodelib/fs.stat/-/fs.stat-2.0.5.tgz", "integrity": "sha512-RkhPPP2zrqDAQA/2jNhnztcPAIv64Xdhlp7a7454A5ovl7Bukxgt7MX7udwAu3zg1DcpPU0rz3VV1SeaqvY4+A==", "dev": true, "license": "MIT", "engines": { "node": ">= 8" }, "node_modules/@nodelib/fs.walk": { "version": "1.2.8", "resolved": "https://registry.npmjs.org/@nodelib/fs.walk/-/fs.walk-1.2.8.tgz", "integrity": "sha512-oGB+UxlgWcgKqgwo8GcEGwemoTfT3FIO9ababBmaGwXlOBKZ+GTy0pP185beGg7Lih/NSHSV2XAs1lnzocSg==", "dev": true, "license": "MIT", "dependencies": { "@nodelib/fs.scandir": "2.1.5", "fastq": "^1.6.0", "engines": { "node": ">= 8" }, "node_modules/@pkgjs/parseargs": { "version": "0.11.0", "resolved": "https://registry.npmjs.org/@pkgjs/parseargs/-/parseargs-0.11.0.tgz", "integrity": "sha512-+1VkjdD0QBLPodGrJUeqarH8VAIvQODlbwh9XpP5Syisf7YoQgsJKPNFoqqLQlu+VQ/tVSshMR6loPMn8U+dPg==", "dev": true, "license": "MIT", "optional": true, "engines": { "node": ">=14" }, "node_modules/@radix-ui/number": { "version": "1.1.1", "resolved": "https://registry.npmjs.org/@radix-ui/number/-/number-1.1.1.tgz", "integrity": "sha512-sha512-MkKCwxIXTgz6CFoJx3pCwn07GKp36+aZyu/u2Ln2VrA5DcdyCZkASEDBTd8x5whTQQL5CiYf4prXKLcgQdv29g==", "license": "MIT", "node_modules/@radix-ui/primitive": { "version": "1.1.3", "resolved": "https://registry.npmjs.org/@radix-ui/primitive/-/primitive-1.1.3.tgz", "integrity": "sha512-JTF99U/6XlJCBo0wqkU5sK10gIYe27MRRsfwoiq5zzOEZLHU3A3KCMa5X/azekYRCJ0HlwI0crAXS/5dEHTzDg==", "license": "MIT", "node_modules/@radix-ui/react-arrow": { "version": "1.1.7", "resolved": "https://registry.npmjs.org/@radix-ui/react-arrow/-/react-arrow-1.1.7.tgz", "integrity": "sha512-F+M1tLhO+mIQAOWspE8Wstg+z6PwxwRd8oQ8IXceWz92kfAmaITRf0EjrrouQeo7QssEPfCn05B4lsh1K9WQ/7w==", "license": "MIT", "dependencies": { "@radix-ui/react-primitive": "2.1.3", "peerDependencies": { "@types/react": "", "@types/react-dom": "" }, "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc", "react-dom": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc", "peerDependenciesMeta": { "@types/react": { "optional": true }, "@types/react-dom": { "optional": true } }, "node_modules/@radix-ui/react-checkbox": { "version": "1.3.3", "resolved":

```

"https://registry.npmjs.org/@radix-ui/react-checkbox/-/react-checkbox-1.3.3.tgz", "integrity": "sha512-wBbpv+NQftHDdG86Qc0plyXk5IR3tM8Vd0nWLKDCx8nNn4nXFOFwsKuqw2okA/1D/mpaAkmuyndrPJTYDNZtFw==", "license": "MIT", "dependencies": { "@radix-ui/primitive": "1.1.3", "@radix-ui/react-compose-refs": "1.1.2", "@radix-ui/react-context": "1.1.2", "@radix-ui/react-presence": "1.1.5", "@radix-ui/react-primitive": "2.1.3", "@radix-ui/react-use-controllable-state": "1.2.2", "@radix-ui/react-use-previous": "1.1.1", "@radix-ui/react-use-size": "1.1.1" }, "peerDependencies": { "@types/react": " ", "@types/react-dom": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc", "react-dom": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true }, "@types/react-dom": { "optional": true } }, "node_modules/@radix-ui/react-collection": { "version": "1.1.7", "resolved": "https://registry.npmjs.org/@radix-ui/react-collection/-/react-collection-1.1.7.tgz", "integrity": "sha512-Fh9rGN0Mol4ZFUNyFVNU4y9LUz93u9/0K+yLgA2bwRojxM8JU1DyvvMBabnZPBgMWREAjvU2jjVzq+LrFUglw==", "license": "MIT", "dependencies": { "@radix-ui/react-compose-refs": "1.1.2", "@radix-ui/react-context": "1.1.2", "@radix-ui/react-primitive": "2.1.3", "@radix-ui/react-slot": "1.2.3" }, "peerDependencies": { "@types/react": " ", "@types/react-dom": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc", "react-dom": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true }, "@types/react-dom": { "optional": true } }, "node_modules/@radix-ui/react-compose-refs": { "version": "1.1.2", "resolved": "https://registry.npmjs.org/@radix-ui/react-compose-refs/-/react-compose-refs-1.1.2.tgz", "integrity": "sha512-z4eqJvfiNnFMHllvXP3CY57y2WJs5g2v3X0zm9mEJkrkNv4rDxu+sg9Jh8EkXyeqBkB7SOcboo9dMVqhyrAClg==", "license": "MIT", "peerDependencies": { "@types/react": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true } }, "node_modules/@radix-ui/react-context": { "version": "1.1.2", "resolved": "https://registry.npmjs.org/@radix-ui/react-context/-/react-context-1.1.2.tgz", "integrity": "sha512-jCi/QKUM2r1Ju5a3J64TH2A5SpKAgh0LpknyqdQ4m6DCV0xJ2HG1xARRwNGPQfi1SLdLWZ1OJz6F4OMBBNiGJA==", "license": "MIT", "peerDependencies": { "@types/react": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true } }, "node_modules/@radix-ui/react-dialog": { "version": "1.1.15", "resolved": "https://registry.npmjs.org/@radix-ui/react-dialog/-/react-dialog-1.1.15.tgz", "integrity": "sha512-TcglVRtZlffRNxRMEyR36DGBLJpeusFcgMVD9PZEzAKnUs1IKCgX5u9BmC2Yg+LL9MgZDugFFs1Vl+Jp4t/PgW==", "license": "MIT", "dependencies": { "@radix-ui/primitive": "1.1.3", "@radix-ui/react-compose-refs": "1.1.2", "@radix-ui/react-context": "1.1.2", "@radix-ui/react-dismissable-layer": "1.1.11", "@radix-ui/react-focus-guards": "1.1.3", "@radix-ui/react-focus-scope": "1.1.7", "@radix-ui/react-id": "1.1.1", "@radix-ui/react-portal": "1.1.9", "@radix-ui/react-presence": "1.1.5", "@radix-ui/react-primitive": "2.1.3", "@radix-ui/react-slot": "1.2.3", "@radix-ui/react-use-controllable-state": "1.2.2", "aria-hidden": "^1.2.4", "react-remove-scroll": "^2.6.3" }, "peerDependencies": { "@types/react": " ", "@types/react-dom": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc", "react-dom": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true }, "@types/react-dom": { "optional": true } }, "node_modules/@radix-ui/react-direction": { "version": "1.1.1", "resolved": "https://registry.npmjs.org/@radix-ui/react-direction/-/react-direction-1.1.1.tgz", "integrity": "sha512-1UEWRX6jnOA2y4H5WczZ44gOOjTEmlqv1uNW4GAJE05+bauCBhv8snY65lw5/VOS/ghKN9gr2KjnLKxrsvoMVw==", "license": "MIT", "peerDependencies": { "@types/react": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true } }, "node_modules/@radix-ui/react-dismissable-layer": { "version": "1.1.11", "resolved": "https://registry.npmjs.org/@radix-ui/react-dismissable-layer/-/react-dismissable-layer-1.1.11.tgz", "integrity": "sha512-Nqcp+t5cTB8BinFkZgXiMjNiQH0PsUt2k51FUhbdfeKvc4ACcG2uQniY/8+h1Yv6Kza4Q7ID7PQV0z0oicE0Mg==", "license": "MIT", "dependencies": { "@radix-ui/primitive": "1.1.3", "@radix-ui/react-compose-refs": "1.1.2", "@radix-ui/react-primitive": "2.1.3", "@radix-ui/react-use-callback-ref": "1.1.1", "@radix-ui/react-use-escape-keydown": "1.1.1" }, "peerDependencies": { "@types/react": " ", "@types/react-dom": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc", "react-dom": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true }, "@types/react-dom": { "optional": true } }, "node_modules/@radix-ui/react-

```

focus-guards": { "version": "1.1.3", "resolved": "https://registry.npmjs.org/@radix-ui/react-focus-guards/-/react-focus-guards-1.1.3.tgz", "integrity": "sha512-0rFg/Rj2Q62NCm62jZw0QX7a3sz6QCQU0LpZdNrJX8byRGaGVTqbrW9jAoIAHyMQqsNpeZ81YgSizOt5WXq0Pw==", "license": "MIT", "peerDependencies": { "@types/react": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true } }, "node_modules/@radix-ui/react-focus-scope": { "version": "1.1.7", "resolved": "https://registry.npmjs.org/@radix-ui/react-focus-scope/-/react-focus-scope-1.1.7.tgz", "integrity": "sha512-t2ODIkXBQyn7jkl6TNaw/MtVEVvIGelJDCG41Okq/KwUsJBwQ4XVZsHAVUkK4mBv3ewiAS3PGuUWuY2BoK4ZUw==", "license": "MIT", "dependencies": { "@radix-ui/react-compose-refs": "1.1.2", "@radix-ui/react-primitive": "2.1.3", "@radix-ui/react-use-callback-ref": "1.1.1" }, "peerDependencies": { "@types/react": " ", "@types/react-dom": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc", "react-dom": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true }, "@types/react-dom": { "optional": true } }, "node_modules/@radix-ui/react-id": { "version": "1.1.1", "resolved": "https://registry.npmjs.org/@radix-ui/react-id/-/react-id-1.1.1.tgz", "integrity": "sha512-kGkGegYldQsOb4XjsfM97rXsiHaBwco+hFI66oO4s9LU+PLAC5oJ7khdOVfXkxsmIbpUqDAvXw11CluXP+jkHg==", "license": "MIT", "dependencies": { "@radix-ui/react-use-layout-effect": "1.1.1" }, "peerDependencies": { "@types/react": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true } }, "node_modules/@radix-ui/react-label": { "version": "2.1.7", "resolved": "https://registry.npmjs.org/@radix-ui/react-label/-/react-label-2.1.7.tgz", "integrity": "sha512-YT1GqPSL8kJn20dJelMX7cTRp/Y9w5IZHvfXQTVHrOqa2yMI7i/UfMqKRU5V7mEyKTrUVgJXhNQPVCG8PBL0Q==", "license": "MIT", "dependencies": { "@radix-ui/react-primitive": "2.1.3" }, "peerDependencies": { "@types/react": " ", "@types/react-dom": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc", "react-dom": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true }, "@types/react-dom": { "optional": true } }, "node_modules/@radix-ui/react-popover": { "version": "1.2.8", "resolved": "https://registry.npmjs.org/@radix-ui/react-popover/-/react-popover-1.2.8.tgz", "integrity": "sha512-0NJQ4LFFUuWkE7Oxf0htBKS6zLkjbBH+hM1uk7Ng705ReR8m/uelduy1DBo0PyBXPkVnBA6YBIU94MBGXrSBCw==", "license": "MIT", "dependencies": { "@floating-ui/react-dom": "2.0.0", "@radix-ui/react-arrow": "1.1.7", "@radix-ui/react-compose-refs": "1.1.2", "@radix-ui/react-context": "1.1.2", "@radix-ui/react-primitive": "2.1.3", "@radix-ui/react-use-callback-ref": "1.1.1", "@radix-ui/react-use-layout-effect": "1.1.1", "@radix-ui/react-use-rect": "1.1.1", "@radix-ui/react-use-size": "1.1.1", "@radix-ui/react": "1.1.1" }, "peerDependencies": { "@types/react": " ", "@types/react-dom": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc", "react-dom": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true }, "@types/react-dom": { "optional": true } }, "node_modules/@radix-ui/react-portal": { "version": "1.1.9", "resolved": "https://registry.npmjs.org/@radix-ui/react-portal/-/react-portal-1.1.9.tgz", "integrity": "sha512-bplxvq03if6UNwXZ+HTK71JLh4APvnXntDc6XOX8UVq4XQOVI7lwok0Avil+b8zgCw3fSaVTZMpAPPagXbKmhQ==", "license": "MIT", "dependencies": { "@radix-ui/react-primitive": "2.1.3", "@radix-ui/react-use-layout-effect": "1.1.1" }, "peerDependencies": { "@types/react": " ", "@types/react-dom": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc", "react-dom": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true }, "@types/react-dom": { "optional": true } }, "node_modules/@radix-ui/react-presence": { "version": "1.1.5", "resolved": "https://registry.npmjs.org/@radix-ui/react-presence/-/react-presence-1.1.5.tgz", "integrity": "sha512-JfEwNDdQVBCNvjKGit4h6pMOzq8bHkopq458dPt2IMjx+eBQUohZNG9A7DtO/O5ukSbxuaNGXMjHicgwy6rQQ==", "license": "MIT", "dependencies": { "@radix-ui/react-compose-refs": "1.1.2", "@radix-ui/react-use-layout-effect": "1.1.1" }, "peerDependencies": { "@types/react": " ", "@types/react-dom": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc", "react-dom": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true }, "@types/react-dom": { "optional": true } }, "node_modules/@radix-ui/react-primitive": { "version": "2.1.3", "resolved": "https://registry.npmjs.org/@radix-ui/react-primitive/-/react-primitive-2.1.3.tgz", "integrity": "sha512-m9gTwRkhy2lvCPe6QJp4d3G1TYEUHn/FzJUtg9MjH46an1wJU+GdoGC5VLof8RX8Ft/DlpshAphkswDLZzHicQ==",

```

```
, "license": "MIT", "dependencies": { "@radix-ui/react-slot": "1.2.3" }, "peerDependencies": { "@types/react": " ",
"@types/react-dom": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc", "react-dom": "^16.8 || ^17.0 || ^18.0 ||
^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true }, "@types/react-dom": {
"optional": true } } }, "node_modules/@radix-ui/react-roving-focus": { "version": "1.1.11", "resolved":
"https://registry.npmjs.org/@radix-ui/react-roving-focus/-/react-roving-focus-1.1.11.tgz", "integrity": "sha512-
7A6S9jSgm/S+7MdtNDSb+IU859vQqJ/QAtcYQcfFC6W8RS4IxIZDldLR0xqCFZ6DCyrQLjLPsxtTNch5jVA4IA==",
"license": "MIT", "dependencies": { "@radix-ui/primitive": "1.1.3", "@radix-ui/react-collection": "1.1.7", "@radix-
ui/react-compose-refs": "1.1.2", "@radix-ui/react-context": "1.1.2", "@radix-ui/react-direction": "1.1.1", "@radix-
ui/react-id": "1.1.1", "@radix-ui/react-primitive": "2.1.3", "@radix-ui/react-use-callback-ref": "1.1.1", "@radix-
ui/react-use-controllable-state": "1.2.2" }, "peerDependencies": { "@types/react": " ", "@types/react-dom": " ",
"react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc", "react-dom": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" },
"peerDependenciesMeta": { "@types/react": { "optional": true }, "@types/react-dom": { "optional": true } } },
"node_modules/@radix-ui/react-select": { "version": "2.2.6", "resolved": "https://registry.npmjs.org/@radix-
ui/react-select/-/react-select-2.2.6.tgz", "integrity": "sha512-
l30RydO+bnn2PQztvo25tswPH+wFBjehVGtmagkU78yMdwTwVf12wnAOF+AeP8S2N8xD+5UPbGhkUfPyvT+mwQ
==", "license": "MIT", "dependencies": { "@radix-ui/number": "1.1.1", "@radix-ui/primitive": "1.1.3", "@radix-ui/react-
collection": "1.1.7", "@radix-ui/react-compose-refs": "1.1.2", "@radix-ui/react-context": "1.1.2", "@radix-ui/react-
direction": "1.1.1", "@radix-ui/react-dismissable-layer": "1.1.11", "@radix-ui/react-focus-guards": "1.1.3", "@radix-
ui/react-focus-scope": "1.1.7", "@radix-ui/react-id": "1.1.1", "@radix-ui/react-popover": "1.2.8", "@radix-ui/react-
portal": "1.1.9", "@radix-ui/react-primitive": "2.1.3", "@radix-ui/react-slot": "1.2.3", "@radix-ui/react-use-callback-
ref": "1.1.1", "@radix-ui/react-use-controllable-state": "1.2.2", "@radix-ui/react-use-layout-effect": "1.1.1", "@radix-
ui/react-use-previous": "1.1.1", "@radix-ui/react-visually-hidden": "1.2.3", "aria-hidden": "^1.2.4", "react-remove-
scroll": "^2.6.3" }, "peerDependencies": { "@types/react": " ", "@types/react-dom": " ", "react": "^16.8 || ^17.0 ||
^18.0 || ^19.0 || ^19.0.0-rc", "react-dom": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": {
"@types/react": { "optional": true }, "@types/react-dom": { "optional": true } } }, "node_modules/@radix-ui/react-
slot": { "version": "1.2.3", "resolved": "https://registry.npmjs.org/@radix-ui/react-slot/-/react-slot-1.2.3.tgz",
"integrity": "sha512-
aeNmHnBxbi2St0au6VBVC7JXFhLlOnVllPNNiyUNAClZmtAUEY8/pBiK3iHjufOlwA+c20/8jngo7xcrg8A==",
"license": "MIT", "dependencies": { "@radix-ui/react-compose-refs": "1.1.2" }, "peerDependencies": {
"@types/react": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": {
"@types/react": { "optional": true } } }, "node_modules/@radix-ui/react-tabs": { "version": "1.1.13", "resolved":
"https://registry.npmjs.org/@radix-ui/react-tabs/-/react-tabs-1.1.13.tgz", "integrity": "sha512-
7xdcatg7/U+7+Udyoj2zodtl9H/llopqo+YOlcZOq1nJwXWBZ9p8xiu5lIXlekDbZkca79a/fozEYQXIA4sW6A==",
"license": "MIT", "dependencies": { "@radix-ui/primitive": "1.1.3", "@radix-ui/react-context": "1.1.2", "@radix-
ui/react-direction": "1.1.1", "@radix-ui/react-id": "1.1.1", "@radix-ui/react-presence": "1.1.5", "@radix-ui/react-
primitive": "2.1.3", "@radix-ui/react-roving-focus": "1.1.11", "@radix-ui/react-use-controllable-state": "1.2.2" },
"peerDependencies": { "@types/react": " ", "@types/react-dom": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 ||
^19.0.0-rc", "react-dom": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": {
"@types/react": { "optional": true }, "@types/react-dom": { "optional": true } } }, "node_modules/@radix-ui/react-
use-callback-ref": { "version": "1.1.1", "resolved": "https://registry.npmjs.org/@radix-ui/react-use-callback-
ref/-/react-use-callback-ref-1.1.1.tgz", "integrity": "sha512-
FkBmWd+qbGQeMu1cOHnuGB6x4yzPjho8ap5WtbEJ26umhgqVXbhekKUQO+hZEL1vU92a3wHwdp0HAcqAUF5i
Dg==", "license": "MIT", "peerDependencies": { "@types/react": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 ||
^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true } } }, "node_modules/@radix-ui/react-
use-controllable-state": { "version": "1.2.2", "resolved": "https://registry.npmjs.org/@radix-ui/react-use-
controllable-state/-/react-use-controllable-state-1.2.2.tgz", "integrity": "sha512-
BjasUjixPFdS+NKkypcyN5Pmg83OlSt0+c6vGov0diwTEo6mgdqvVR6hxcEgFuh4QrAs7Rc+9KuGJ9TVCj0Zzg==",
"license": "MIT", "dependencies": { "@radix-ui/react-use-effect-event": "0.0.2", "@radix-ui/react-use-layout-
effect": "1.1.1" }, "peerDependencies": { "@types/react": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" },
```

```

"peerDependenciesMeta": { "@types/react": { "optional": true } } }, "node_modules/@radix-ui/react-use-effect-event": { "version": "0.0.2", "resolved": "https://registry.npmjs.org/@radix-ui/react-use-effect-event/-/react-use-effect-event-0.0.2.tgz", "integrity": "sha512-Qp8WbZOB+blgpuUT+lw2xheLP8q0oatc9UpmiemEICxGvFLYmHm9QowVZGHtJIGbS6A6yJ3iViad/2cVjnOiA==", "license": "MIT", "dependencies": { "@radix-ui/react-use-layout-effect": "1.1.1" }, "peerDependencies": { "@types/react": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true } } }, "node_modules/@radix-ui/react-use-escape-keydown": { "version": "1.1.1", "resolved": "https://registry.npmjs.org/@radix-ui/react-use-escape-keydown/-/react-use-escape-keydown-1.1.1.tgz", "integrity": "sha512-ll0+boE7w/XebUHyBjroE+DbByORGR9KKmITzbR7MyQ4akpORYP/ZmbhAr0DG7RmmBqoOnZdy2QlVajJ2QA59g==", "license": "MIT", "dependencies": { "@radix-ui/react-use-callback-ref": "1.1.1" }, "peerDependencies": { "@types/react": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true } } }, "node_modules/@radix-ui/react-use-layout-effect": { "version": "1.1.1", "resolved": "https://registry.npmjs.org/@radix-ui/react-use-layout-effect/-/react-use-layout-effect-1.1.1.tgz", "integrity": "sha512-RbJRS4UWQFkzHTTwwVymMTUv8EqYhOp8dOOviLj2ugtTiXRaRQS7GLGxZTLL1jWhMeoSCf5zmcZkqTl9liYfXcQ==", "license": "MIT", "peerDependencies": { "@types/react": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true } } }, "node_modules/@radix-ui/react-use-previous": { "version": "1.1.1", "resolved": "https://registry.npmjs.org/@radix-ui/react-use-previous/-/react-use-previous-1.1.1.tgz", "integrity": "sha512-2dHfToCj/pzca2Ck724OZ5L0EVrr3eHRNsG/b3xQJLA2hZpVCS99bLAX+hm1IHxDenzU6by5z/5MIY794/a8NQ==", "license": "MIT", "peerDependencies": { "@types/react": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true } } }, "node_modules/@radix-ui/react-use-rect": { "version": "1.1.1", "resolved": "https://registry.npmjs.org/@radix-ui/react-use-rect/-/react-use-rect-1.1.1.tgz", "integrity": "sha512-QTYuDesS0VtuHNNvMh+CjIKJ4LJickCMUAqjIE3+j8w+RIRpwyX3apEQKGFzbZGdo7XNG1tXa+bQqIE7HIXT2w==", "license": "MIT", "dependencies": { "@radix-ui/rect": "1.1.1" }, "peerDependencies": { "@types/react": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true } } }, "node_modules/@radix-ui/react-use-size": { "version": "1.1.1", "resolved": "https://registry.npmjs.org/@radix-ui/react-use-size/-/react-use-size-1.1.1.tgz", "integrity": "sha512-ewrXRDTAqAXIkI6t/fkXWNAhFX9I+CkKlw6zjEwk86RSPKwZr3xpBRso655aqYafwtbnbpHLj6toFzmd6xdVptQ==", "license": "MIT", "dependencies": { "@radix-ui/react-use-layout-effect": "1.1.1" }, "peerDependencies": { "@types/react": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true } } }, "node_modules/@radix-ui/react-visually-hidden": { "version": "1.2.3", "resolved": "https://registry.npmjs.org/@radix-ui/react-visually-hidden/-/react-visually-hidden-1.2.3.tgz", "integrity": "sha512-pzJq12tEaalhqbzpcuv/OypJY/BPavOofm+dbab+MHLajy277+1lM6JFcGgF5eskJ6mquGirhXY2GD/8u8Ug==", "license": "MIT", "dependencies": { "@radix-ui/react-primitive": "2.1.3" }, "peerDependencies": { "@types/react": " ", "@types/react-dom": " ", "react": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc", "react-dom": "^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc" }, "peerDependenciesMeta": { "@types/react": { "optional": true }, "@types/react-dom": { "optional": true } } }, "node_modules/@radix-ui/rect": { "version": "1.1.1", "resolved": "https://registry.npmjs.org/@radix-ui/rect/-/rect-1.1.1.tgz", "integrity": "sha512-HPwPGlzk128mWyZqG52jqDJ12waP11Pa1lGoiyUklEuMLBP0oeK/C89esbXrxsky5we7dfd8U58nm0SgAWpVw==", "license": "MIT" }, "node_modules/@remix-run/router": { "version": "1.23.0", "resolved": "https://registry.npmjs.org/@remix-run/router/-/router-1.23.0.tgz", "integrity": "sha512-O3rHJzAQKamUz1fvE0Qaw0xSFqsA/yafi2iqeE0pvdFtCO1viYx8QL6f3Ln/aCCTLxs68SLf0KPM9eSeM8yBnA==", "engines": { "node": ">=14.0.0" }, "node_modules/@rolldown/pluginutils": { "version": "1.0.0-beta.35", "resolved": "https://registry.npmjs.org/@rolldown/pluginutils/-/pluginutils-1.0.0-beta.35.tgz", "integrity": "sha512-slYrCpoxJUqzFDDNlvrOYRazQUNRvWPjXA17dAOISY3rDMxX6k8K4cj2H+hEYMHF81HO3uNd5rHVigAWRM5dSg=

```

```
=, "dev": true, "license": "MIT" }, "node_modules/@rollup/rollup-android-arm-eabi": { "version": "4.50.2",  
"resolved": "https://registry.npmjs.org/@rollup/rollup-android-arm-eabi/-/rollup-android-arm-eabi-4.50.2.tgz",  
"integrity": "sha512-  
uLN8NAiFVIRKX9ZQha8wy6UUs06UNSZ32xj6giK/rmMXAgKahwExvK6SsmgU5/brh4w/nSgj8e0k3c1HBQpa0A==",  
"cpu": [ "arm" ], "dev": true, "license": "MIT", "optional": true, "os": [ "android" ] },  
"node_modules/@rollup/rollup-android-arm64": { "version": "4.50.2", "resolved":  
"https://registry.npmjs.org/@rollup/rollup-android-arm64/-/rollup-android-arm64-4.50.2.tgz", "integrity":  
"sha512-  
oEouqQk2/zxxj22PNcGSskya+3kV0ZKH+nQxuCCOGJ4oTXBdNTbv+f/E3c74cNLeMO1S5wVWacSws10TTSB77g=  
=", "cpu": [ "arm64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "android" ] },  
"node_modules/@rollup/rollup-darwin-arm64": { "version": "4.50.2", "resolved":  
"https://registry.npmjs.org/@rollup/rollup-darwin-arm64/-/rollup-darwin-arm64-4.50.2.tgz", "integrity": "sha512-  
OZuTVTpj3CDSlXmPgGH8en/XtirV5nfljHZ3wrNwvgkT5DQLhLKAEuFSiwtbMto6oVexV0k1F1zqURPKf5r1lQ==",  
"cpu": [ "arm64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "darwin" ] },  
"node_modules/@rollup/rollup-darwin-x64": { "version": "4.50.2", "resolved":  
"https://registry.npmjs.org/@rollup/rollup-darwin-x64/-/rollup-darwin-x64-4.50.2.tgz", "integrity": "sha512-  
Wa/Wn8RFklr1vy1k1PB/VYhLnInn5eaJkftQKivirOvzu5uVd2lt01ukeQstMursuz7S1bU+8WW+1UPXpa8A==",  
"cpu": [ "x64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "darwin" ] }, "node_modules/@rollup/rollup-  
freebsd-arm64": { "version": "4.50.2", "resolved": "https://registry.npmjs.org/@rollup/rollup-freebsd-  
arm64/-/rollup-freebsd-arm64-4.50.2.tgz", "integrity": "sha512-  
QkzxvH3kYN9J1w7D1A+yIMdl1pPekD+pWx7G5rXgnlIQ1TVYVC6hLI7SOV9pi5q9ulDF9AulGkuzcbF7+fAhow==",  
"cpu": [ "arm64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "freebsd" ] },  
"node_modules/@rollup/rollup-freebsd-x64": { "version": "4.50.2", "resolved":  
"https://registry.npmjs.org/@rollup/rollup-freebsd-x64/-/rollup-freebsd-x64-4.50.2.tgz", "integrity": "sha512-  
dkYXB0c2XAS3a3jmyDkX4Jk0m7gWLFzq1C3qUnJJ38AyXlF5G/dyS4N9B30nvFseCfqtCEdbYFhk0ChoCGxPog==",  
"cpu": [ "x64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "freebsd" ] },  
"node_modules/@rollup/rollup-linux-arm-gnueabi": { "version": "4.50.2", "resolved":  
"https://registry.npmjs.org/@rollup/rollup-linux-arm-gnueabi/-/rollup-linux-arm-gnueabi-4.50.2.tgz",  
"integrity": "sha512-  
9VIPY/BN3AgbukfVHAB8zNFWB/IKEuvzRo1NKEv0Po8sYFKx0i+AQICyftgEjcl43F2h9Ui1ZSdVBc4En/sP2w==",  
"cpu": [ "arm" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ] }, "node_modules/@rollup/rollup-  
linux-arm-musleabi": { "version": "4.50.2", "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-arm-  
musleabi/-/rollup-linux-arm-musleabi-4.50.2.tgz", "integrity": "sha512-  
+GdKWOvsifaYNIIVf07QYan1J5F141+vGm5/Y8b9uCZnG/nxOGqGcmR24mv0koIWWuqvFYnbURRqw1lv7IBINw==",  
"cpu": [ "arm" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ] }, "node_modules/@rollup/rollup-  
linux-arm64-gnu": { "version": "4.50.2", "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-arm64-  
gnu/-/rollup-linux-arm64-gnu-4.50.2.tgz", "integrity": "sha512-  
df0Eou14ojtUdLQdPFnymEQteENwSJAdLf5KCDrmZNSy1c3YaCNaJvYsEUHnrg+/DLBH612/R0xd3dD03uz2dg==",  
"cpu": [ "arm64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ] }, "node_modules/@rollup/rollup-  
linux-arm64-musl": { "version": "4.50.2", "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-arm64-  
musl/-/rollup-linux-arm64-musl-4.50.2.tgz", "integrity": "sha512-  
iPeouV0UIDtz8j1YFR4OJ/zf7evjauqv7jQ/EFs0CllY+by++hiaDAfFipjOgyz6y6xbDvJuiU4HwpVMpRFDQ==", "cpu": [  
"arm64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ] }, "node_modules/@rollup/rollup-linux-  
loong64-gnu": { "version": "4.50.2", "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-loong64-  
gnu/-/rollup-linux-loong64-gnu-4.50.2.tgz", "integrity": "sha512-  
OL6KaNVbopLj5fTa5D5bau4W82f+1TyTZrR2BdnfsrnQnmdxh4okMxR2DcDkjuh4KeoQZVuvHvzuD/lyLn2Kw==",  
"cpu": [ "loong64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ] },  
"node_modules/@rollup/rollup-linux-ppc64-gnu": { "version": "4.50.2", "resolved":  
"https://registry.npmjs.org/@rollup/rollup-linux-ppc64-gnu/-/rollup-linux-ppc64-gnu-4.50.2.tgz", "integrity":
```

```
"sha512-
I21VJI1w6z/K5OTRI6aS9DDsgezEZ/yKpbqlvfHbW0CEF5IL8ATBMuUx6/mp683rKTK8thjs/0BaNrZLXetLag==",
"cpu": [ "ppc64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ] }, "node_modules/@rollup/rollup-
linux-riscv64-gnu": { "version": "4.50.2", "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-riscv64-
gnu/-/rollup-linux-riscv64-gnu-4.50.2.tgz", "integrity": "sha512-
Hq6aQJT/qFFHrYmJS20nV+9SKrXL2lvFBENZoKfoTH2kKDOJqff5OSJr4x72ZaG/uUn+XmBnGhfr4lwMRmqCQ==",
"cpu": [ "riscv64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ] },
"node_modules/@rollup/rollup-linux-riscv64-musl": { "version": "4.50.2", "resolved":
"https://registry.npmjs.org/@rollup/rollup-linux-riscv64-musl/-/rollup-linux-riscv64-musl-4.50.2.tgz", "integrity":
"sha512-
82rBSEXRv5qtKyr0xZ/YMF531oj2AlpLZkeNYxmKNN6I2sVE9PGegN99tYDLK2fYHJITL1P2Lgb4ZXnv0PjQvw==",
"cpu": [ "riscv64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ] },
"node_modules/@rollup/rollup-linux-s390x-gnu": { "version": "4.50.2", "resolved":
"https://registry.npmjs.org/@rollup/rollup-linux-s390x-gnu/-/rollup-linux-s390x-gnu-4.50.2.tgz", "integrity":
"sha512-
4Q3S3Hy7pC6uaRo9gtXUTJ+EKO9AKs3BXKc2jYypEcMQ49gDPFU2P1ariX9SEtBzE5eglX6fSUmbmGazwBVF9w=
=", "cpu": [ "s390x" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ] },
"node_modules/@rollup/rollup-linux-x64-gnu": { "version": "4.50.2", "resolved":
"https://registry.npmjs.org/@rollup/rollup-linux-x64-gnu/-/rollup-linux-x64-gnu-4.50.2.tgz", "integrity": "sha512-
9Jie/At6qk70dNlCopcL4p+1UirusEtznpNtcq/u/C5cC4HBX7qSGsYIcG6bdxj15EYWhHiu02YvmdPzylZIA==", "cpu":
[ "x64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ] }, "node_modules/@rollup/rollup-linux-
x64-musl": { "version": "4.50.2", "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-x64-musl/-/rollup-
linux-x64-musl-4.50.2.tgz", "integrity": "sha512-
HPNJwxPL3EmhzeAnsWQCM3DcoqOz3/IC6de9rWfGR8ZCuEHETi9km66bH/wG3YH0V3nyzyFEGUZeL5PKyy4xv
w==", "cpu": [ "x64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "linux" ] },
"node_modules/@rollup/rollup-openharmony-arm64": { "version": "4.50.2", "resolved":
"https://registry.npmjs.org/@rollup/rollup-openharmony-arm64/-/rollup-openharmony-arm64-4.50.2.tgz",
"integrity": "sha512-
nMKvq6FRHSzYfKLHZ+cChowIEkR2lj/V0jYj9JnGUVPL2/mleFGmVM2mLaFeNa5Jev7W7TovXqXIG2d39y1KYA==",
"cpu": [ "arm64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "openharmony" ] },
"node_modules/@rollup/rollup-win32-arm64-msvc": { "version": "4.50.2", "resolved":
"https://registry.npmjs.org/@rollup/rollup-win32-arm64-msvc/-/rollup-win32-arm64-msvc-4.50.2.tgz", "integrity":
"sha512-
eFUvvnTYEKeTyHEijQKz81bLrUQOXKZqECeiWH6tb8eXXbZk+CXSG2aFrig2BQ/pjiVRj36zysjgILkqarS2YA==",
"cpu": [ "arm64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "win32" ] },
"node_modules/@rollup/rollup-win32-ia32-msvc": { "version": "4.50.2", "resolved":
"https://registry.npmjs.org/@rollup/rollup-win32-ia32-msvc/-/rollup-win32-ia32-msvc-4.50.2.tgz", "integrity":
"sha512-
cBaWmXqyfRhH8zmUxK3d3sAhEWLrtMjWBRwdMMHJIXSjvJLkv49adxiEz+FJ8AP90apSDDbX2Tyd/WyIV6ikA==",
"cpu": [ "ia32" ], "dev": true, "license": "MIT", "optional": true, "os": [ "win32" ] }, "node_modules/@rollup/rollup-
win32-x64-msvc": { "version": "4.50.2", "resolved": "https://registry.npmjs.org/@rollup/rollup-win32-x64-
msvc/-/rollup-win32-x64-msvc-4.50.2.tgz", "integrity": "sha512-
APwKy6YUvhZaEoHyM+9xqmTpvIEl+9eL7LoCH+aLcvWYHJ663qG5zx7WzWZY+a9qkg5JtzcMyJ9z0WtQBMDmg
A==", "cpu": [ "x64" ], "dev": true, "license": "MIT", "optional": true, "os": [ "win32" ] },
"node_modules/@types/babel__core": { "version": "7.20.5", "resolved":
"https://registry.npmjs.org/@types/babel__core/-/babel__core-7.20.5.tgz", "integrity": "sha512-
qoQprZvz5wQFJwMDqeseRXWv3rqMvhgpbXFfVYWhbx9X47POIA6i+dXefEmZKoAgOaTdalgnSMqMIU61yRyzA=
=", "dev": true, "license": "MIT", "dependencies": { "@babel/parser": "^7.20.7", "@babel/types": "^7.20.7",
"@types/babel__generator": " ", "@types/babel__template": " ", "@types/babel__traverse": " " } },
```



```
"node_modules/@types/babel__generator": { "version": "7.27.0", "resolved":  
"https://registry.npmjs.org/@types/babel__generator/-/babel__generator-7.27.0.tgz", "integrity": "sha512-  
ufFd2Xi92OAVPYsy+P4n7/U7e68fex0+Ee8gSG9KX7eo084CWiq4sdxtvdl0bOPupXtVJPY19zk6EwWqUQ8lg==",  
"dev": true, "license": "MIT", "dependencies": { "@babel/types": "^7.0.0" } },  
"node_modules/@types/babel__template": { "version": "7.4.4", "resolved":  
"https://registry.npmjs.org/@types/babel__template/-/babel__template-7.4.4.tgz", "integrity": "sha512-  
h/NUaSyG5EyxBlp8YRxo4RMe2/qQgvyowRwVMzhYhBCONbW8PUsg4lkFMrhgZhUe5z3L3MiLDuvyJ/CaPa2A8A=  
=", "dev": true, "license": "MIT", "dependencies": { "@babel/parser": "^7.1.0", "@babel/types": "^7.0.0" } },  
"node_modules/@types/babel__traverse": { "version": "7.28.0", "resolved":  
"https://registry.npmjs.org/@types/babel__traverse/-/babel__traverse-7.28.0.tgz", "integrity": "sha512-  
8PvcXf70gTDZBgt9ptxJ8elBeBjcLOAcOtoO/mPJtji1+CdGbHgm77om1GrSPxiE+uXlpNSK64UYalwQXd4Q==",  
"dev": true, "license": "MIT", "dependencies": { "@babel/types": "^7.28.2" } }, "node_modules/@types/estree": {  
"version": "1.0.8", "resolved": "https://registry.npmjs.org/@types/estree/-/estree-1.0.8.tgz", "integrity": "sha512-  
dWHzHa2WqEXl/O1E9OjrocMTKJl2mSrEolh1lomrv6U+JuNwaHXsXx9bLu5gG7BUWFIN0sklQJQ/L1rlex4X6w==",  
"dev": true, "license": "MIT" }, "node_modules/@types/json-schema": { "version": "7.0.15", "resolved":  
"https://registry.npmjs.org/@types/json-schema/-/json-schema-7.0.15.tgz", "integrity": "sha512-  
5+fP8P8MFNC+AyZCDxrB2pkZFPgzqQWUzpSeuuVLvm8VMcorNYavBqoFcXK8bQz4Qsbn4oUEEm4wDLfcysGH  
A==", "dev": true, "license": "MIT" }, "node_modules/@types/react": { "version": "19.1.13", "resolved":  
"https://registry.npmjs.org/@types/react/-/react-19.1.13.tgz", "integrity": "sha512-  
hHkbU/eoO3EG5/MZkuFSKmYqPbSVk5byPFa3e7y/8TybHiLMACgl8seVYlicwk7H5K/rI2px9xrQp/C+AUDTiQ==",  
"devOptional": true, "license": "MIT", "dependencies": { "csstype": "^3.0.2" } }, "node_modules/@types/react-  
dom": { "version": "19.1.9", "resolved": "https://registry.npmjs.org/@types/react-dom/-/react-dom-19.1.9.tgz",  
"integrity": "sha512-  
qXRuZaOsAdXKFyOhRBg6Lqqc0yay13vN7Krlg4L7N4aaHN68ma9OK3NE1BoDFgFOTfM7zg+3/8+2n8rLUH3OKQ  
==", "devOptional": true, "license": "MIT", "peerDependencies": { "@types/react": "^19.0.0" } },  
"node_modules/@vitejs/plugin-react": { "version": "5.0.3", "resolved": "https://registry.npmjs.org/@vitejs/plugin-  
react/-/plugin-react-5.0.3.tgz", "integrity": "sha512-  
PFVHHosKkofGH0Yzrw1BipSedTH68BFF8ZWY1kfUpCtJcouXXY0+racG8sExw7hw0HoX36813ga5o3LTWZ4FUg=  
=", "dev": true, "license": "MIT", "dependencies": { "@babel/core": "^7.28.4", "@babel/plugin-transform-react-jsx-  
self": "^7.27.1", "@babel/plugin-transform-react-jsx-source": "^7.27.1", "@rollup/pluginutils": "1.0.0-beta.35",  
"@types/babel__core": "^7.20.5", "react-refresh": "^0.17.0" }, "engines": { "node": "^20.19.0 || >=22.12.0" },  
"peerDependencies": { "vite": "^4.2.0 || ^5.0.0 || ^6.0.0 || ^7.0.0" } }, "node_modules/acorn": { "version": "8.15.0",  
"resolved": "https://registry.npmjs.org/acorn/-/acorn-8.15.0.tgz", "integrity": "sha512-  
NZyJarBfL7nWwlq+FDL6Zp/yHEhePMNnnJ0y3qfieCrmNvYct8uvTV41UvLSe6apAfk0fY1FbWx+NwfmptvTg==",  
"dev": true, "license": "MIT", "bin": { "acorn": "bin/acorn" }, "engines": { "node": ">=0.4.0" } },  
"node_modules/acorn-jsx": { "version": "5.3.2", "resolved": "https://registry.npmjs.org/acorn-jsx/-/acorn-jsx-  
5.3.2.tgz", "integrity": "sha512-  
rq9s+JNhfoIChjtDXxllJ7g41oZk5SIXtp0LHwyA5cejwn7vKmKp4pPri6YEePv2PU65sAsegbXtlinmDFDXgQ==",  
"dev": true, "license": "MIT", "peerDependencies": { "acorn": "^6.0.0 || ^7.0.0 || ^8.0.0" } }, "node_modules/ajv": {  
"version": "6.12.6", "resolved": "https://registry.npmjs.org/ajv/-/ajv-6.12.6.tgz", "integrity": "sha512-  
j3fVLgvTo527anyYyJOGTYJbG+vnQYvE0m5mmkc1TK+nxAppkCLMIL0aZ4dbIVCNoGShhm+kzE4ZUykBoMg4g=  
=", "dev": true, "license": "MIT", "dependencies": { "fast-deep-equal": "^3.1.1", "fast-json-stable-stringify": "^2.0.0",  
"json-schema-traverse": "^0.4.1", "uri-js": "^4.2.2" }, "funding": { "type": "github", "url":  
"https://github.com/sponsors/epoberezkin" } }, "node_modules/ansi-regex": { "version": "6.2.2", "resolved":  
"https://registry.npmjs.org/ansi-regex/-/ansi-regex-6.2.2.tgz", "integrity": "sha512-  
Bq3SmSpyFHAwJpK8If9yc6svM8c56dB5BAW4Qbw5jHTwwXXcTLORMkpDJp6VL0XzIWaCHTXrkFURMYmD0sLq  
g==", "dev": true, "license": "MIT", "engines": { "node": ">=12" }, "funding": { "url": "https://github.com/chalk/ansi-  
regex?sponsor=1" } }, "node_modules/ansi-styles": { "version": "4.3.0", "resolved": "https://registry.npmjs.org/ansi-  
styles/-/ansi-styles-4.3.0.tgz", "integrity": "sha512-  
"
```

```
zbB9rCJAT1rbjiVDdb2hqKFHNYLxgk8NURxZ3IZwD3F6NtxbXZQCnnSi1Lkx+IDohdPIFP222wVALIheZJQSEg==",
"dev": true, "license": "MIT", "dependencies": { "color-convert": "^2.0.1" }, "engines": { "node": ">=8" }, "funding": {
"url": "https://github.com/chalk/ansi-styles?sponsor=1" } }, "node_modules/any-promise": { "version": "1.3.0",
"resolved": "https://registry.npmjs.org/any-promise/-/any-promise-1.3.0.tgz", "integrity": "sha512-7UvmKalWRt1wgjL1RrGxoSJW/0QZFlegpeGvZG9kjp8vrRu55XTHbwnqq2GpXm9uLbcuhxm3lqX9OB4MZR1b2A=
=", "dev": true, "license": "MIT" }, "node_modules/anymatch": { "version": "3.1.3", "resolved":
"https://registry.npmjs.org/anymatch/-/anymatch-3.1.3.tgz", "integrity": "sha512-7UvmKalWRt1wgjL1RrGxoSJW/0QZFlegpeGvZG9kjp8vrRu55XTHbwnqq2GpXm9uLbcuhxm3lqX9OB4MZR1b2A=
=", "dev": true, "license": "ISC", "dependencies": { "normalize-path": "^3.0.0", "picomatch": "^2.0.4" }, "engines": {
"node": ">= 8" } }, "node_modules/anymatch/node_modules/picomatch": { "version": "2.3.1", "resolved":
"https://registry.npmjs.org/picomatch/-/picomatch-2.3.1.tgz", "integrity": "sha512-7UvmKalWRt1wgjL1RrGxoSJW/0QZFlegpeGvZG9kjp8vrRu55XTHbwnqq2GpXm9uLbcuhxm3lqX9OB4MZR1b2A=
=", "dev": true, "license": "MIT", "engines": { "node": ">=8.6" }, "funding": { "url":
"https://github.com/sponsors/jonschlinkert" } }, "node_modules/arg": { "version": "5.0.2", "resolved":
"https://registry.npmjs.org/arg/-/arg-5.0.2.tgz", "integrity": "sha512-7UvmKalWRt1wgjL1RrGxoSJW/0QZFlegpeGvZG9kjp8vrRu55XTHbwnqq2GpXm9uLbcuhxm3lqX9OB4MZR1b2A=
=", "dev": true, "license": "MIT" }, "node_modules/argparse": { "version": "2.0.1", "resolved":
"https://registry.npmjs.org/argparse/-/argparse-2.0.1.tgz", "integrity": "sha512-7UvmKalWRt1wgjL1RrGxoSJW/0QZFlegpeGvZG9kjp8vrRu55XTHbwnqq2GpXm9uLbcuhxm3lqX9OB4MZR1b2A=
8+9WqebbfZpX9OR+Wa6O29aslogeRMzcGtAlNdpMHHyAg10f05aSFVBbcEqGf/PXw1EjAZ+q2/bEBg3DvurK3Q=
=", "dev": true, "license": "Python-2.0" }, "node_modules/aria-hidden": { "version": "1.2.6", "resolved":
"https://registry.npmjs.org/aria-hidden/-/aria-hidden-1.2.6.tgz", "integrity": "sha512-7UvmKalWRt1wgjL1RrGxoSJW/0QZFlegpeGvZG9kjp8vrRu55XTHbwnqq2GpXm9uLbcuhxm3lqX9OB4MZR1b2A=
ik3ZgC9dY/IYVVM++OISsaYDeg1tb0VtP5uL3ouh1koGOaUMDPpbFlei4JkFimWUFPn90sbMNMXQAIVOLnYKJA==",
"license": "MIT", "dependencies": { "tslib": "^2.0.0" }, "engines": { "node": ">=10" } }, "node_modules/autoprefixer":
{ "version": "10.4.21", "resolved": "https://registry.npmjs.org/autoprefixer/-/autoprefixer-10.4.21.tgz", "integrity":
"sha512-7UvmKalWRt1wgjL1RrGxoSJW/0QZFlegpeGvZG9kjp8vrRu55XTHbwnqq2GpXm9uLbcuhxm3lqX9OB4MZR1b2A=
O+A6LWV5LDHSD3LjHYoNi4VLsj/Whi7k6zG12xTYaU4cQ8oxQGckXNX8cRHK5yOZ/ppVHe0ZBXGzSV9jXdVbQ
=", "dev": true, "funding": [ { "type": "opencollective", "url": "https://opencollective.com/postcss/" }, { "type":
"tidelift", "url": "https://tidelift.com/funding/github/npm/autoprefixer" }, { "type": "github", "url":
"https://github.com/sponsors/ai" } ], "license": "MIT", "dependencies": { "browserslist": "^4.24.4", "caniuse-lite":
"^1.0.30001702", "fraction.js": "^4.3.7", "normalize-range": "^0.1.2", "picocolors": "^1.1.1", "postcss-value-parser":
"^4.2.0" }, "bin": { "autoprefixer": "bin/autoprefixer" }, "engines": { "node": "^10 || ^12 || >=14" },
"peerDependencies": { "postcss": "^8.1.0" }, "node_modules/balanced-match": { "version": "1.0.2", "resolved":
"https://registry.npmjs.org/balanced-match/-/balanced-match-1.0.2.tgz", "integrity": "sha512-7UvmKalWRt1wgjL1RrGxoSJW/0QZFlegpeGvZG9kjp8vrRu55XTHbwnqq2GpXm9uLbcuhxm3lqX9OB4MZR1b2A=
3oSeUO0TMV67hN1AmbXsK4yaqU7tjiHlbrRDZOpH0KW9+CeX4bRAaX0Anxt0tx2MrpRpWwQaPwllISEJhYU5Pw=
=", "dev": true, "license": "MIT" }, "node_modules/baseline-browser-mapping": { "version": "2.8.6", "resolved":
"https://registry.npmjs.org/baseline-browser-mapping/-/baseline-browser-mapping-2.8.6.tgz", "integrity":
"sha512-7UvmKalWRt1wgjL1RrGxoSJW/0QZFlegpeGvZG9kjp8vrRu55XTHbwnqq2GpXm9uLbcuhxm3lqX9OB4MZR1b2A=
wrH5NNqren/QMtKUEEJf7z86YjfqW/2uw3IL3/xpqZUC95SSVIFXYQeeGjL6FT/X68IROu6RMehZQS5foy2BXw==",
"dev": true, "license": "Apache-2.0", "bin": { "baseline-browser-mapping": "dist/cli.js" }, "node_modules/binary-
extensions": { "version": "2.3.0", "resolved": "https://registry.npmjs.org/binary-extensions/-/binary-extensions-
2.3.0.tgz", "integrity": "sha512-7UvmKalWRt1wgjL1RrGxoSJW/0QZFlegpeGvZG9kjp8vrRu55XTHbwnqq2GpXm9uLbcuhxm3lqX9OB4MZR1b2A=
Ceh+7ox5qe7LJuLHoY0feh3pHuUDHAcRUeyL2VYghZwfpkNly/+8Ocg0a3UuSoYzavmylwuLWQOf3hl0jjMMlw==",
"dev": true, "license": "MIT", "engines": { "node": ">=8" }, "funding": { "url":
"https://github.com/sponsors/sindresorhus" } }, "node_modules/brace-expansion": { "version": "1.1.12", "resolved":
"https://registry.npmjs.org/brace-expansion/-/brace-expansion-1.1.12.tgz", "integrity": "sha512-7UvmKalWRt1wgjL1RrGxoSJW/0QZFlegpeGvZG9kjp8vrRu55XTHbwnqq2GpXm9uLbcuhxm3lqX9OB4MZR1b2A=
9T9UjW3r0UW5c1Q7GTWllPtXwhvYmEzFhzMfZ9H7FQWt+uZePjZPjBP/W1ZEyZ1twGWom5/56TF4IPcqjndHcg==",
"dev": true, "license": "MIT", "dependencies": { "balanced-match": "^1.0.0", "concat-map": "0.0.1" } },
"node_modules/braces": { "version": "3.0.3", "resolved": "https://registry.npmjs.org/braces/-/braces-3.0.3.tgz",
"integrity": "sha512-7UvmKalWRt1wgjL1RrGxoSJW/0QZFlegpeGvZG9kjp8vrRu55XTHbwnqq2GpXm9uLbcuhxm3lqX9OB4MZR1b2A=
```

yQbXgO/OSZVD2IsiLlro+7Hf6Q18EJrKSEsdoMzKePKXct3gvD8oLcOQdlzGupr5Fj+EDe8gO/lxc1BzfMpxvA==",
"dev": true, "license": "MIT", "dependencies": { "fill-range": "^7.1.1" }, "engines": { "node": ">=8" } },
"node_modules/browserslist": { "version": "4.26.2", "resolved":
"https://registry.npmjs.org/browserslist/-/browserslist-4.26.2.tgz", "integrity": "sha512-
ECFzp6uFOSB+dcZ5BK/IBaGwssbSYBHvuMeMt3MMFyhI0Z8SqGgEkBLARgpRH3hutlgPVsALcMwbDrJqPxQ65A
==", "dev": true, "funding": [{ "type": "opencollective", "url": "https://opencollective.com/browserslist" }, { "type":
"tidelift", "url": "https://tidelift.com/funding/github/npm/browserslist" }, { "type": "github", "url":
"https://github.com/sponsors/ai" }], "license": "MIT", "dependencies": { "baseline-browser-mapping": "^2.8.3",
"caniuse-lite": "^1.0.30001741", "electron-to-chromium": "^1.5.218", "node-releases": "^2.0.21", "update-
browserslist-db": "^1.1.3" }, "bin": { "browserslist": "cli.js" }, "engines": { "node": "^6 || ^7 || ^8 || ^9 || ^10 || ^11 ||
^12 || >=13.7" } }, "node_modules/callsites": { "version": "3.1.0", "resolved":
"https://registry.npmjs.org/callsites/-/callsites-3.1.0.tgz", "integrity": "sha512-
P8BjAsXvZS+VIDUI11hHCQEv74YT67YUi5JJFNWIqL235sBmjX4+qx9Muvls5ivyNENctx46xQLQ3aTuE7ssaQ==",
"dev": true, "license": "MIT", "engines": { "node": ">=6" } }, "node_modules/camelcase-css": { "version": "2.0.1",
"resolved": "https://registry.npmjs.org/camelcase-css/-/camelcase-css-2.0.1.tgz", "integrity": "sha512-
QOSvevhsljgYwRx6Rv7zKdMF8lbRmx+uQGx2+vDc+KI/eBnsy9kit5aj23AgGu3pa4t9AgwbnXWqS+iOY+2aA==",
"dev": true, "license": "MIT", "engines": { "node": ">= 6" } }, "node_modules/caniuse-lite": { "version":
"1.0.30001743", "resolved": "https://registry.npmjs.org/caniuse-lite/-/caniuse-lite-1.0.30001743.tgz", "integrity":
"sha512-
e6Ojr7RV14Un7dz6ASD0aZDmQPT/A+eZU+nuTNfjqmRrmkmQInTNWH0SKmqagx9PeW87UVqapSurtAXifmtdmw
==", "dev": true, "funding": [{ "type": "opencollective", "url": "https://opencollective.com/browserslist" }, { "type":
"tidelift", "url": "https://tidelift.com/funding/github/npm/caniuse-lite" }, { "type": "github", "url":
"https://github.com/sponsors/ai" }], "license": "CC-BY-4.0" }, "node_modules/chalk": { "version": "4.1.2",
"resolved": "https://registry.npmjs.org/chalk/-/chalk-4.1.2.tgz", "integrity": "sha512-
oKnbfyRiXpUueez8iBmMyEa4nbj4IOQyuhc/wy9kY7/WVPcwIO9VA668Pu8RkO7+0G76SLROeyw9CpQ061i4mA=
==", "dev": true, "license": "MIT", "dependencies": { "ansi-styles": "^4.1.0", "supports-color": "^7.1.0" }, "engines": {
"node": ">=10" }, "funding": { "url": "https://github.com/chalk/chalk?sponsor=1" } }, "node_modules/chokidar": {
"version": "3.6.0", "resolved": "https://registry.npmjs.org/chokidar/-/chokidar-3.6.0.tgz", "integrity": "sha512-
7VT13fmjotKpGipCW9JEQAusEPE+Ei8nl6/g4FBAmIm0GOOLMua9NDDo/DWp0ZAxCr3cPq5ZpBqmPAQgDda2Pw
==", "dev": true, "license": "MIT", "dependencies": { "anymatch": "~3.1.2", "braces": "~3.0.2", "glob-parent":
"~5.1.2", "is-binary-path": "~2.1.0", "is-glob": "~4.0.1", "normalize-path": "~3.0.0", "readdirp": "~3.6.0" }, "engines":
{ "node": ">= 8.10.0" }, "funding": { "url": "https://paulmillr.com/funding/" }, "optionalDependencies": { "fsevents":
"~2.3.2" } }, "node_modules/chokidar/node_modules/glob-parent": { "version": "5.1.2", "resolved":
"https://registry.npmjs.org/glob-parent/-/glob-parent-5.1.2.tgz", "integrity": "sha512-
AOlgSQCEpiJYwP3ARnGx+5VnTu2HBYdzBGP45eLw1vr3zB3vZLeyed1sC9hnbcOc9/SrMyM5RPQrkGz4aS9Zow==
", "dev": true, "license": "ISC", "dependencies": { "is-glob": "^4.0.1" }, "engines": { "node": ">= 6" } },
"node_modules/class-variance-authority": { "version": "0.7.1", "resolved": "https://registry.npmjs.org/class-
variance-authority/-/class-variance-authority-0.7.1.tgz", "integrity": "sha512-
Ka+9Trutv7G8M6WT6SeiRWz792K5qEqIGEGzXKhAE6xOWAY6pPH8U+9IY3oCMv6kqTmLsv7Xh/2w2RigkePMsg
==", "license": "Apache-2.0", "dependencies": { "clsx": "^2.1.1" }, "funding": { "url": "https://polar.sh/cva" } },
"node_modules/clsx": { "version": "2.1.1", "resolved": "https://registry.npmjs.org/clsx/-/clsx-2.1.1.tgz", "integrity":
"sha512-
"sha512-
eYm0QWBtUrBWZWG0d386OGAw16Z995PiOVo2B7bjWSbHedGI5e0ZWaq65kOGgUSNesEIDkBI9SbTg/JK9dhCZ
A==", "license": "MIT", "engines": { "node": ">=6" } }, "node_modules/color-convert": { "version": "2.0.1",
"resolved": "https://registry.npmjs.org/color-convert/-/color-convert-2.0.1.tgz", "integrity": "sha512-
RRECPsj7iu/xb5oKYcsFHSppFNnsj/52OVTRKb4zP5onXwVF3zVmmToNcOfGC+CRDpfK/U584fMg38ZHCaElKQ==
", "dev": true, "license": "MIT", "dependencies": { "color-name": "~1.1.4" }, "engines": { "node": ">=7.0.0" } },
"node_modules/color-name": { "version": "1.1.4", "resolved": "https://registry.npmjs.org/color-name/-/color-name-
1.1.4.tgz", "integrity": "sha512-
67 / 108

dOy+3AuW3a2wNbZHluMZpTcgjGuLU/uBL/ubcZF9OXbDo8ff4O8yVp5Bf0efS8uEoYo5q4Fx7dY9OgQGxGAsQA=
=", "dev": true, "license": "MIT", "node_modules/commander": { "version": "4.1.1", "resolved":
"https://registry.npmjs.org/commander/-/commander-4.1.1.tgz", "integrity": "sha512-NOKm8xhkzAjzFx8B2v5OAHT+u5pRQc2UCa2Vq9jYL/31o2wi9mxBA7LIFs3sV5VSC49z6pEhfbMULvShKj26WA==
", "dev": true, "license": "MIT", "engines": { "node": ">= 6" } }, "node_modules/concat-map": { "version": "0.0.1",
"resolved": "https://registry.npmjs.org/concat-map/-/concat-map-0.0.1.tgz", "integrity":
"sha512-Srv4dswyQNBfohGpz9o6Yb3Gz3SrUDqBH5rTuhGR7ahtlbYKnVxw2bCFMRljaA7EXHaXZ8wsHdodFvbkh
Kmqg==", "dev": true, "license": "MIT", "node_modules/convert-source-map": { "version": "2.0.0", "resolved":
"https://registry.npmjs.org/convert-source-map/-/convert-source-map-2.0.0.tgz", "integrity": "sha512-Kvp459HrV2FEJ1CA5i1Ku+MY3kasH19TFyKtZ2xWmMeq6bk2NU3XXvfJ+Q61m0xktWwt+1HSYf3JZsTms3aRJg==
", "dev": true, "license": "MIT", "node_modules/cross-spawn": { "version": "7.0.6", "resolved":
"https://registry.npmjs.org/cross-spawn/-/cross-spawn-7.0.6.tgz", "integrity": "sha512-uV2QOWP2nWzsy2aMp8aRibhi9dlzF5Hgh5SHaB9OiTGEyDTiJJyx0uy51QXdyWbtAHNua4XJzUKca3OzKUd3vA=
=", "dev": true, "license": "MIT", "dependencies": { "path-key": "^3.1.0", "shebang-command": "^2.0.0", "which":
"^2.0.1" }, "engines": { "node": ">= 8" } }, "node_modules/cssesc": { "version": "3.0.0", "resolved":
"https://registry.npmjs.org/cssesc/-/cssesc-3.0.0.tgz", "integrity":
"sha512-/Tb/Jcjk111nNScGob5MNtsntNM1aCNUdipB/TkwZFhyDrrE47SOx/18wF2bbjgc3ZzCSKW1T5nt5EbFoAz/
Vg==", "dev": true, "license": "MIT", "bin": { "cssesc": "bin/cssesc" }, "engines": { "node": ">=4" } },
"node_modules/csstype": { "version": "3.1.3", "resolved": "https://registry.npmjs.org/csstype/-/csstype-3.1.3.tgz",
"integrity": "sha512-M1uQkMI8rQK/szD0LNhtqXlPLpimGm8sOBwU7lLnCpSbTyY3yeU1Vc7l4KT5zT4s/yOxHH5O7tluuLOCnLADRW==",
"devOptional": true, "license": "MIT", "node_modules/debug": { "version": "4.4.3", "resolved":
"https://registry.npmjs.org/debug/-/debug-4.4.3.tgz", "integrity": "sha512-
RGwwWnwQvkVfavKVt22FGLw+xYSdzARwm0ru6DhTVA3umU5hZc28V3kO4stgYrrTILpuvgl9GijltAjNbcqA==",
"dev": true, "license": "MIT", "dependencies": { "ms": "^2.1.3" }, "engines": { "node": ">=6.0" },
"peerDependenciesMeta": { "supports-color": { "optional": true } } }, "node_modules/deep-is": { "version": "0.1.4",
"resolved": "https://registry.npmjs.org/deep-is/-/deep-is-0.1.4.tgz", "integrity": "sha512-
oIPzksmTg4/MriiaYGO+okXDT7ztn/w3Eptv/+gSldMdKsJo0u4CfYNFJPy+4SKMuCqGw2wxnA+URMg3t8a/bQ==",
"dev": true, "license": "MIT", "node_modules/detect-node-es": { "version": "1.1.0", "resolved":
"https://registry.npmjs.org/detect-node-es/-/detect-node-es-1.1.0.tgz", "integrity": "sha512-
ypdmJU/TbBby2Dxibuv7ZLW3Bs1QEmM7nHjEANfohJLvE0XVujisn1qPJcZxg+qDucs+rBp6fLD1rPS3AhJ7EQ==",
"license": "MIT", "node_modules/didyoumean": { "version": "1.2.2", "resolved":
"https://registry.npmjs.org/didyoumean/-/didyoumean-1.2.2.tgz", "integrity": "sha512-
gxytqMg7GKyhQmb056K7M3xszy/myH8w+B4RT+QXBQsvAOdc3XymqDDPHx1BgPgSDAA5Slifona89YtRATDzw=
=", "dev": true, "license": "Apache-2.0", "node_modules/dlv": { "version": "1.1.3", "resolved":
"https://registry.npmjs.org/dlv/-/dlv-1.1.3.tgz", "integrity": "sha512-
+HlytjIPKniG8XuRG8WvmBP8xs8P71y+SKKS6ZXWoEgLuePxtDoUEiH7WkdePWrrQ5JBpE6aoVqfZfJUQkjXwA==",
"dev": true, "license": "MIT", "node_modules/eastasianwidth": { "version": "0.2.0", "resolved":
"https://registry.npmjs.org/eastasianwidth/-/eastasianwidth-0.2.0.tgz", "integrity": "sha512-
l88TYZWc9XiYHRQ4/3c5rjfgkjhLyW2luGIheGERbNQ6OY7yTybanSpDXa8y7VUP9YmDcYa+eyq4ca7iLqWA==",
"dev": true, "license": "MIT", "node_modules/electron-to-chromium": { "version": "1.5.221", "resolved":
"https://registry.npmjs.org/electron-to-chromium/-/electron-to-chromium-1.5.221.tgz", "integrity":
"sha512-1hFJ39wkW01ogqSyYoA4goOXOtMRy6B+yvA1u42nnsEGtHzlzmK93aPiSumVQebJl47JUHLc9coCjUxb1E
vtQ==", "dev": true, "license": "ISC", "node_modules/emoji-regex": { "version": "9.2.2", "resolved":
"https://registry.npmjs.org/emoji-regex/-/emoji-regex-9.2.2.tgz", "integrity": "sha512-
L18DaJsXSUK2+42pv8mLs5jJT2hqFkFE4j21wOmgbUqsZ2hL72NsUU785g9RXgo3s0ZNgVI42TiHp3ZtOv/Vyg==",
"dev": true, "license": "MIT", "node_modules/esbuild": { "version": "0.25.10", "resolved":
"https://registry.npmjs.org/esbuild/-/esbuild-0.25.10.tgz", "integrity": "sha512-
9RiGKvCwaqxO2owP61uQ4BgNborAQskMR6QusfWzQqv7AZOg5oGehdY2pRJMTKuwx1IDBP4rSbl5IHZU7SMsQ

```

==, "dev": true, "hasInstallScript": true, "license": "MIT", "bin": { "esbuild": "bin/esbuild" }, "engines": { "node":
">=18" }, "optionalDependencies": { "@esbuild/aix-ppc64": "0.25.10", "@esbuild/android-arm": "0.25.10",
"@esbuild/android-arm64": "0.25.10", "@esbuild/android-x64": "0.25.10", "@esbuild/darwin-arm64": "0.25.10",
"@esbuild/darwin-x64": "0.25.10", "@esbuild/freebsd-arm64": "0.25.10", "@esbuild/freebsd-x64": "0.25.10",
"@esbuild/linux-arm": "0.25.10", "@esbuild/linux-arm64": "0.25.10", "@esbuild/linux-ia32": "0.25.10",
"@esbuild/linux-loong64": "0.25.10", "@esbuild/linux-mips64el": "0.25.10", "@esbuild/linux-ppc64": "0.25.10",
"@esbuild/linux-riscv64": "0.25.10", "@esbuild/linux-s390x": "0.25.10", "@esbuild/linux-x64": "0.25.10",
"@esbuild/netbsd-arm64": "0.25.10", "@esbuild/netbsd-x64": "0.25.10", "@esbuild/openbsd-arm64": "0.25.10",
"@esbuild/openbsd-x64": "0.25.10", "@esbuild/openharmony-arm64": "0.25.10", "@esbuild/sunos-x64": "0.25.10",
"@esbuild/win32-arm64": "0.25.10", "@esbuild/win32-ia32": "0.25.10", "@esbuild/win32-x64": "0.25.10" },
"node_modules/escalade": { "version": "3.2.0", "resolved": "https://registry.npmjs.org/escalade/-/escalade-
3.2.0.tgz", "integrity": "sha512-
WUj2qlxaQtO4g6Pq5c29GTcWGdYd8itL8zTlipgECz3JesAiiOKotd8JU6otB3PACgG6xkJUyVhboMS+bje/jA==",
"dev": true, "license": "MIT", "engines": { "node": ">=6" }, "node_modules/escape-string-regexp": { "version":
"4.0.0", "resolved": "https://registry.npmjs.org/escape-string-regexp/-/escape-string-regexp-4.0.0.tgz",
"integrity": "sha512-
TtpcNJ3XAZx3Gq8sWRzJaVajRs0uVxA2YAkdb1jm2YkPz4G6egUFAyA3n5vtEIzefPk5Wa4UXbKuS5fKkJWdgA==",
"dev": true, "license": "MIT", "engines": { "node": ">=10" }, "funding": { "url":
"https://github.com/sponsors/sindresorhus" }, "node_modules/eslint": { "version": "9.35.0", "resolved":
"https://registry.npmjs.org/eslint/-/eslint-9.35.0.tgz", "integrity": "sha512-
QePbBFMJFjgmIE+cXAlbHZbHpdFVS2E/6vzCy7aKlebddv1vadiC4JFV5u/wqTkNUwEV8WrQi257jf5f06hrg==",
"dev": true, "license": "MIT", "dependencies": { "@eslint-community/eslint-utils": "^4.8.0", "@eslint-
community/regexpp": "^4.12.1", "@eslint/config-array": "^0.21.0", "@eslint/config-helpers": "^0.3.1", "@eslint/core":
"^0.15.2", "@eslint/eslintrc": "^3.3.1", "@eslint/js": "9.35.0", "@eslint/plugin-kit": "^0.3.5", "@humanfs/node":
"^0.16.6", "@humanwhocodes/module-importer": "^1.0.1", "@humanwhocodes/retry": "^0.4.2", "@types/estree":
"^1.0.6", "@types/json-schema": "^7.0.15", "ajv": "^6.12.4", "chalk": "^4.0.0", "cross-spawn": "^7.0.6", "debug":
"^4.3.2", "escape-string-regexp": "^4.0.0", "eslint-scope": "^8.4.0", "eslint-visitor-keys": "^4.2.1", "esprez":
"^10.4.0", "esquery": "^1.5.0", "esutils": "^2.0.2", "fast-deep-equal": "^3.1.3", "file-entry-cache": "^8.0.0", "find-up":
"^5.0.0", "glob-parent": "^6.0.2", "ignore": "^5.2.0", "imurmurhash": "^0.1.4", "is-glob": "^4.0.0", "json-stable-
stringify-without-jsonify": "^1.0.1", "lodash.merge": "^4.6.2", "minimatch": "^3.1.2", "natural-compare": "^1.4.0",
"optionator": "^0.9.3" }, "bin": { "eslint": "bin/eslint.js" }, "engines": { "node": "^18.18.0 || ^20.9.0 || >=21.1.0" },
"funding": { "url": "https://eslint.org/donate" }, "peerDependencies": { "jiti": " " }, "peerDependenciesMeta": { "jiti":
{ "optional": true } }, "node_modules/eslint-plugin-react-hooks": { "version": "5.2.0", "resolved":
"https://registry.npmjs.org/eslint-plugin-react-hooks/-/eslint-plugin-react-hooks-5.2.0.tgz", "integrity": "sha512-
+f15FfK64YQwZdJNELETdn5ibXEUQmW1DZL6KXhNnc2heoy/sg9VJJt7n8TIMWouzWqSWavFklhHylblAEapg==",
"dev": true, "license": "MIT", "engines": { "node": ">=10" }, "peerDependencies": { "eslint": "^3.0.0 || ^4.0.0 ||
^5.0.0 || ^6.0.0 || ^7.0.0 || ^8.0.0-0 || ^9.0.0" }, "node_modules/eslint-plugin-react-refresh": { "version": "0.4.20",
"resolved": "https://registry.npmjs.org/eslint-plugin-react-refresh/-/eslint-plugin-react-refresh-0.4.20.tgz",
"integrity": "sha512-
XpbHQ2q5gUF8BGOX4dHe+71qoirYMhApEPZ7sfhF/dNnOF1UXnCMGZf79SFTBO7Bz5YEIT4TMieSIJBWhP9WBA
==, "dev": true, "license": "MIT", "peerDependencies": { "eslint": ">=8.40" }, "node_modules/eslint-scope": {
"version": "8.4.0", "resolved": "https://registry.npmjs.org/eslint-scope/-/eslint-scope-8.4.0.tgz", "integrity":
"sha512-
sNXOfKCn74rt8RICKMvJS7XKV/Xk9kA7DyJr8mJik3S7Cwgy3qlkkmyS2uQB3jiJg6VNdzd/pDBJu0nvG2NITg==",
"dev": true, "license": "BSD-2-Clause", "dependencies": { "esrecurse": "^4.3.0", "estraverse": "^5.2.0" },
"engines": { "node": "^18.18.0 || ^20.9.0 || >=21.1.0" }, "funding": { "url": "https://opencollective.com/eslint" },
"node_modules/eslint-visitor-keys": { "version": "4.2.1", "resolved": "https://registry.npmjs.org/eslint-visitor-
keys/-/eslint-visitor-keys-4.2.1.tgz", "integrity": "sha512-
UhdK5sfqcee/9H/rCOJikYz67o0a2Tw2hGRPOG2Y1R2dg7brRe1uG0yaNQDHu+TO/uQPF/5eCapvYSmHUjt7JQ==",

```

```
"dev": true, "license": "Apache-2.0", "engines": { "node": "^18.18.0 || ^20.9.0 || >=21.1.0" }, "funding": { "url":  
"https://opencollective.com/eslint" } }, "node_modules/esprex": { "version": "10.4.0", "resolved":  
"https://registry.npmjs.org/esprex/-/esprex-10.4.0.tgz", "integrity": "sha512-  
j6PAQ2uUr79PZhBjP5C5fhl8e39FmRnOjsD5lGnWrfU8i2G776tBK7+nP8KuQUTTyAZUwfQqXAgVH5MbH9CYQ=  
=", "dev": true, "license": "BSD-2-Clause", "dependencies": { "acorn": "^8.15.0", "acorn-jsx": "^5.3.2", "eslint-  
visitor-keys": "^4.2.1" }, "engines": { "node": "^18.18.0 || ^20.9.0 || >=21.1.0" }, "funding": { "url":  
"https://opencollective.com/eslint" } }, "node_modules/esquery": { "version": "1.6.0", "resolved":  
"https://registry.npmjs.org/esquery/-/esquery-1.6.0.tgz", "integrity": "sha512-  
ca9pw9fomFcKPvFLXhBKUK90ZvGibiGOvRJNbjlY7s7uq/5YO4BOzcYtJqExdx99rF6aAcnRxHmcUHcz6sQsg==",  
"dev": true, "license": "BSD-3-Clause", "dependencies": { "estraverse": "^5.1.0" }, "engines": { "node": ">=0.10" }  
}, "node_modules/esrecurse": { "version": "4.3.0", "resolved": "https://registry.npmjs.org/esrecurse/-/esrecurse-  
4.3.0.tgz", "integrity": "sha512-  
KmfKL3b6G+RXvP8N1vr3Tq1kL/oCFgn2NYXEtqP8/L3pKapUA4G8cFVaoF3SU323CD4XypR/ffioHmkti6/Tag==",  
"dev": true, "license": "BSD-2-Clause", "dependencies": { "estraverse": "^5.2.0" }, "engines": { "node": ">=4.0" }  
}, "node_modules/estraverse": { "version": "5.3.0", "resolved": "https://registry.npmjs.org/estraverse/-/estraverse-  
5.3.0.tgz", "integrity": "sha512-  
MMdARuVEQziNTEJD8DgMqmhWR11BRQ/cBP+pLTYdSTnf3MIO8fFeilNEbX36ZdNfU/7A9f3gUw49B3oQsvwBA==",  
"dev": true, "license": "BSD-2-Clause", "engines": { "node": ">=4.0" } }, "node_modules/esutils": { "version":  
"2.0.3", "resolved": "https://registry.npmjs.org/esutils/-/esutils-2.0.3.tgz", "integrity": "sha512-  
kVscqXk4OCp68SZ0dkgEKVi6/8ij300KBWTJq32P/dYeWTSwK41WyTxaIN1eRmA5Z9UU/LX9D7FWSmV9SAYx6g=  
=", "dev": true, "license": "BSD-2-Clause", "engines": { "node": ">=0.10.0" } }, "node_modules/fast-deep-equal": {  
"version": "3.1.3", "resolved": "https://registry.npmjs.org/fast-deep-equal/-/fast-deep-equal-3.1.3.tgz", "integrity":  
"sha512-  
f3qQ9oQy9j2AhBe/H9VC91wLmKBCCU/gDOnKNAYG5hswO7BLKj09Hc5HYNz9cGI++xlpDCIgDaitVs03ATR84Q=  
=", "dev": true, "license": "MIT" }, "node_modules/fast-glob": { "version": "3.3.3", "resolved":  
"https://registry.npmjs.org/fast-glob/-/fast-glob-3.3.3.tgz", "integrity": "sha512-  
7MptL8U0cqcfDzIzwOTHoIX9x5BrNqye7Z/LuC7kCMRio1EMSyqRK3BEAUD7sXRq4iT4AzTVuZdhgQ2TCvYlg==",  
"dev": true, "license": "MIT", "dependencies": { "@nodelib/fs.stat": "^2.0.2", "@nodelib/fs.walk": "^1.2.3", "glob-  
parent": "^5.1.2", "merge2": "^1.3.0", "micromatch": "^4.0.8" }, "engines": { "node": ">=8.6.0" } },  
"node_modules/fast-glob/node_modules/glob-parent": { "version": "5.1.2", "resolved":  
"https://registry.npmjs.org/glob-parent/-/glob-parent-5.1.2.tgz", "integrity": "sha512-  
AOlgSQcepijYwP3ARnGx+5VnTu2HBYdzbGP45eLw1vr3zB3vZLeyed1sC9hnbcOc9/SrMyM5RPQrkGz4aS9Zow==  
", "dev": true, "license": "ISC", "dependencies": { "is-glob": "^4.0.1" }, "engines": { "node": ">= 6" } },  
"node_modules/fast-json-stable-stringify": { "version": "2.1.0", "resolved": "https://registry.npmjs.org/fast-json-  
stable-stringify/-/fast-json-stable-stringify-2.1.0.tgz", "integrity": "sha512-  
lhd/wF+Lk98HZoTCTiVraHthf5XYijjalXck7saUtuanSDyLMxnHhSXEDJqHxD7msR8D0uCmqlkwjCV8xvwHw==",  
"dev": true, "license": "MIT" }, "node_modules/fast-levenshtein": { "version": "2.0.6", "resolved":  
"https://registry.npmjs.org/fast-levenshtein/-/fast-levenshtein-2.0.6.tgz", "integrity": "sha512-  
DCXu6lfhqqcs7TZKY3Hxp3y6qphY5SJZmrWMDrKcERSOXWQdMhU9Ig/PYrzyw/ul9jOlyh0N4M0tbC5hodg8dw==  
", "dev": true, "license": "MIT" }, "node_modules/fastq": { "version": "1.19.1", "resolved":  
"https://registry.npmjs.org/fastq/-/fastq-1.19.1.tgz", "integrity": "sha512-  
GwLTyxkCXjXbxqlhTsMI2Nui8huMPtnxg7krajPJAjnEG/iiOS7i+CztWGR9G0NBKbXKh6X9m9UIsYX/N6vvQ==",  
"dev": true, "license": "ISC", "dependencies": { "reusify": "^1.0.4" } }, "node_modules/fdir": { "version": "6.5.0",  
"resolved": "https://registry.npmjs.org/fdir/-/fdir-6.5.0.tgz", "integrity": "sha512-  
tlbYtZbucOs0BRGqPjKshJUYdL+SDH7dVVM8ggy+ERp3WAUJLEFJE+02kanyHtwjWOnwrKYBiwAmM0p4kLJAnXg==  
", "dev": true, "license": "MIT", "engines": { "node": ">=12.0.0" }, "peerDependencies": { "picomatch": "^3 || ^4",  
"peerDependenciesMeta": { "picomatch": { "optional": true } } }, "node_modules/file-entry-cache": { "version":  
"8.0.0", "resolved": "https://registry.npmjs.org/file-entry-cache/-/file-entry-cache-8.0.0.tgz", "integrity": "sha512-  
XXTUwCvisa5oacNGRP9SfNtYBNAMi+RPwBFmbIZEF7N7swHYQS6/Zfk7SRwx4D5j3CH211YNRco1DEMNVfZCnQ
```

```

==, "dev": true, "license": "MIT", "dependencies": { "flat-cache": "^4.0.0" }, "engines": { "node": ">=16.0.0" } },
"node_modules/fill-range": { "version": "7.1.1", "resolved": "https://registry.npmjs.org/fill-range/-/fill-range-7.1.1.tgz", "integrity": "sha512-78837874001237d48565d43b2277d0348c5c78e792d82164ad96174439416f7", "dev": true, "license": "MIT", "dependencies": { "to-regex-range": "^5.0.1" }, "engines": { "node": ">=8" } },
"node_modules/find-up": { "version": "5.0.0", "resolved": "https://registry.npmjs.org/find-up/-/find-up-5.0.0.tgz", "integrity": "sha512-18A0hJ1GDFKRL2O2/wgnwLgC1gjwNROU3v3hNN0D2Z7wXkU8/qv7sKq4O0L1hE9XN/60N3BguDnHt4pYYA==",
78/PXT1wILLDgTzDs7sjq9hzz0vXD+zn+7wypEe4fXQxCmdmqfGsEPQxmiCSQl3ajFV91bVSsvNtrJRiW6nGng==",
"dev": true, "license": "MIT", "dependencies": { "locate-path": "^6.0.0", "path-exists": "^4.0.0" }, "engines": {
"node": ">=10" }, "funding": { "url": "https://github.com/sponsors/sindresorhus" } }, "node_modules/flat-cache": {
"version": "4.0.1", "resolved": "https://registry.npmjs.org/flat-cache/-/flat-cache-4.0.1.tgz", "integrity": "sha512-3QVUHDO0CxeO8Q86Jnt1bd5pFqZ3W+1wZXHkqBZVL3FipYt8Y4oU9YK6W3qP81Z6N1L7330yD9881331A==",
"dev": true, "license": "MIT", "dependencies": { "flatted": "^3.2.9", "keyv": "^4.5.4" }, "engines": { "node": ">=16"
} }, "node_modules/flatted": { "version": "3.3.3", "resolved": "https://registry.npmjs.org/flatted/-/flatted-3.3.3.tgz",
"integrity": "sha512-1W21Z2Ux02AOXXGcx77q80WQe0Lz8zoK1xxtR9OD0aTieDqSxqj993mHvzA0s0Y4NkK8UHwY7H21H8pQ==",
GX+ysw4PBCz0PzosHDepZGANEuFCMLrnRTiEy9McGjmkCQYwRq4A/X786G/fjM/+OjsWSU1ZrY5qyARZmO/uwg=
==, "dev": true, "license": "ISC" }, "node_modules/foreground-child": { "version": "3.3.1", "resolved":
"https://registry.npmjs.org/foreground-child/-/foreground-child-3.3.1.tgz", "integrity": "sha512-ycF2W0QkiYR4yaWwL4bc2ky1yQW25N9aVnJYm81wqPOHk7FVW4bRwI7aH95V65aM1v78+XzUxO6rYDg==",
"dev": true, "license": "ISC", "dependencies": { "cross-spawn": "^7.0.6", "signal-exit": "^4.0.1" }, "engines": {
"node": ">=14" }, "funding": { "url": "https://github.com/sponsors/isaacs" } }, "node_modules/fraction.js": {
"version": "4.3.7", "resolved": "https://registry.npmjs.org/fraction.js/-/fraction.js-4.3.7.tgz", "integrity": "sha512-5X8zR7wX5Dd8hROu3u958zSX59L01aKXWU88Hq30G8+7VbK1F0KqF0sWwX097Cn07gkxvRW5G5/47A==",
"dev": true, "license": "MIT", "engines": { "node": " " }, "funding": { "type": "patreon", "url":
"https://github.com/sponsors/rawify" } }, "node_modules/fsevents": { "version": "2.3.3", "resolved":
"https://registry.npmjs.org/fsevents/-/fsevents-2.3.3.tgz", "integrity": "sha512-9a1e9z50mZ9D9919A3fS5Vf5iKUH1i1J8yVq0wn+bKOCuKw7vP+UwL47nXzEPoW1xmVrYUzqU8mV8fRQ==",
5xoDfX+fl7faATnagmWPpbFtwh/R77WmMMqqHGS65C3vvB0YHrgF+B1YmZ3441tMj5n63k0212XNoJwzlhffQw=
==, "dev": true, "hasInstallScript": true, "license": "MIT", "optional": true, "os": [ "darwin" ], "engines": { "node":
"^8.16.0 || ^10.6.0 || >=11.0.0" } }, "node_modules/function-bind": { "version": "1.1.2", "resolved":
"https://registry.npmjs.org/function-bind/-/function-bind-1.1.2.tgz", "integrity": "sha512-7P8xKkDDREzovvHJX0/GtN3rlkQm99XWIF9/S7ffHVAOQD+JLZSvnuKrku5Y/EVWS5Uz3I+jdsQ6uZcM36zAA==",
7XHNxH7qX9xG5mlwxkhumTox/MIRNcOgDrxWsMt2pAr23WHP6MrRIN7FBSFpCpr+oVO0F744iUgR82nJMfG2SA
==, "dev": true, "license": "MIT", "funding": { "url": "https://github.com/sponsors/ljharb" } },
"node_modules/gensync": { "version": "1.0.0-beta.2", "resolved": "https://registry.npmjs.org/gensync/-/gensync-1.0.0-beta.2.tgz", "integrity": "sha512-3hN7NaskYvMDLQY55gnW3NQ+mesEAepTqlg+VEbj7zzqEMBVNhzcGYEeqFo/TIYz6eQiFcp1HcsCZO+nGgS8zg=
==, "dev": true, "license": "MIT", "engines": { "node": ">=6.9.0" } }, "node_modules/get-nonce": { "version": "1.0.1",
"resolved": "https://registry.npmjs.org/get-nonce/-/get-nonce-1.0.1.tgz", "integrity": "sha512-ocuz48b26I1W7/pfKaUuN77mnYqUzF/6GutwP6DBE3hmWxbtE5HGpMXfY8FgfHw784757591a+38YUQ==",
FJhYRoDaiatfEKUK8HKlicmu/3SGFD51q3itKDGOSTysQJBnfOcxU5GxnhE1E6soB76MbT0MBtnKJuXyAx+96Q==",
"license": "MIT", "engines": { "node": ">=6" } }, "node_modules/glob": { "version": "10.4.5", "resolved":
"https://registry.npmjs.org/glob/-/glob-10.4.5.tgz", "integrity": "sha512-7BvRrF0k6xjo7d4A/PxYLBUCfb6c+Vpd2/mB2yRDlew7Jb5hEXiCD9ibfO7wpk8i4sevK6DFny9h7EYbM3/sHg==",
"dev": true, "license": "ISC", "dependencies": { "foreground-child": "^3.1.0", "jackspeak": "^3.1.2", "minimatch":
"^9.0.4", "minipass": "^7.1.2", "package-json-from-dist": "^1.0.0", "path-scurry": "^1.11.1" }, "bin": { "glob":
"dist/esm/bin.mjs" }, "funding": { "url": "https://github.com/sponsors/isaacs" } }, "node_modules/glob-parent": {
"version": "6.0.2", "resolved": "https://registry.npmjs.org/glob-parent/-/glob-parent-6.0.2.tgz", "integrity":
"sha512-1xv0394KbT4S41677fF7s36r/Ow9K34sVhEsLepDJ9AqK36EgiLW2EPxBVxqEO3I46hGtfwcuKQh6LH8g==",
sha512-1sha512-
XxwI8EOhVQgWp6iDL+3b0r86f4d6AX6zSU55HfB4ydCEuXLXc5FcYeOu+nnGftS4TEju/11rt4KJPTMgbfmv4A==",
"dev": true, "license": "ISC", "dependencies": { "is-glob": "^4.0.3" }, "engines": { "node": ">=10.13.0" } },
"node_modules/glob/node_modules/brace-expansion": { "version": "2.0.2", "resolved":

```

```

"https://registry.npmjs.org/brace-expansion/-/brace-expansion-2.0.2.tgz", "integrity": "sha512-Jt0vHyM+ jmUBqojB7E1N1Yadt0vi0Qxjxd2TErW94wDz+E2LAm5vKMXXwg6ZZBTHPuUIDgQHkXvjGBdfcF1ZDQ=
=", "dev": true, "license": "MIT", "dependencies": { "balanced-match": "^1.0.0" } },
"node_modules/glob/node_modules/minimatch": { "version": "9.0.5", "resolved":
"https://registry.npmjs.org/minimatch/-/minimatch-9.0.5.tgz", "integrity": "sha512-
G6T0ZX48xgozx7587koeX9Ys2NYy6Gmv//P89sEte9V9whlapMNF4idKxnW2QtCcLiTWlb/wfCabAtAFWhhBow==",
, "dev": true, "license": "ISC", "dependencies": { "brace-expansion": "^2.0.1" }, "engines": { "node": ">=16 || 14
>=14.17" }, "funding": { "url": "https://github.com/sponsors/isaacs" } }, "node_modules/globals": { "version":
"16.4.0", "resolved": "https://registry.npmjs.org/globals/-/globals-16.4.0.tgz", "integrity": "sha512-
ob/2LcVVaVGCYN+r14cnwnoDPuUfjiYgSqRhiFD0Q1il4Odora5RE8lv1D24hAz5oMophRGkGz+yuvQmmUMnMw==",
, "dev": true, "license": "MIT", "engines": { "node": ">=18" }, "funding": { "url":
"https://github.com/sponsors/sindresorhus" } }, "node_modules/has-flag": { "version": "4.0.0", "resolved":
"https://registry.npmjs.org/has-flag/-/has-flag-4.0.0.tgz", "integrity": "sha512-
EykJT/Q1KjTWctppgIAgfSO0tKVuZUjhgMr17kqTumMl6Afv3EISleU7qZUzoXDFTAHTDC4NOoG/ZxU3EvlMPQ==",
"dev": true, "license": "MIT", "engines": { "node": ">=8" } }, "node_modules/hasown": { "version": "2.0.2",
"resolved": "https://registry.npmjs.org/hasown/-/hasown-2.0.2.tgz", "integrity": "sha512-
0hJU9SCPvmMzIBdZFqNPXWa6dqh7WdH0cII9y+Cys8rG3nL48Bclra9HmKhVVUHyPWNH5Y7xDwAB7bfgSjkUM
Q==", "dev": true, "license": "MIT", "dependencies": { "function-bind": "^1.1.2" }, "engines": { "node": ">= 0.4" } },
"node_modules/ignore": { "version": "5.3.2", "resolved": "https://registry.npmjs.org/ignore/-/ignore-5.3.2.tgz",
"integrity": "sha512-
hsBTNUqQTDwkWtcdYI2i06Y/nUBEsNEDJKjWdigLvegy8kDuJAS8uRlpkkcQpyEXL0Z/pjDy5HBmMjRCJ2gq+g==",
"dev": true, "license": "MIT", "engines": { "node": ">= 4" } }, "node_modules/import-fresh": { "version": "3.3.1",
"resolved": "https://registry.npmjs.org/import-fresh/-/import-fresh-3.3.1.tgz", "integrity": "sha512-
TR3KfrTZTYLPB6jUjfx6MF9WcWrHL9su5TOBK4ZkYgBdWKPOFoSoQldEuTuR82pmtxH2spWG9h6etwfr1pLBqQ=
=", "dev": true, "license": "MIT", "dependencies": { "parent-module": "^1.0.0", "resolve-from": "^4.0.0" }, "engines":
{ "node": ">=6" }, "funding": { "url": "https://github.com/sponsors/sindresorhus" } },
"node_modules/imurmurhash": { "version": "0.1.4", "resolved":
"https://registry.npmjs.org/imurmurhash/-/imurmurhash-0.1.4.tgz", "integrity": "sha512-
JmXmZ6wuvDmLiHEml9ykzqO6lWfbof0GG4lkcGaEndCRDDmMVnny7s5HsIgHCbaq0w2MyPhDqkhTUgS2LU2PH
A==", "dev": true, "license": "MIT", "engines": { "node": ">=0.8.19" } }, "node_modules/is-binary-path": { "version":
"2.1.0", "resolved": "https://registry.npmjs.org/is-binary-path/-/is-binary-path-2.1.0.tgz", "integrity": "sha512-
ZMERYes6pDydyuGidse7OsHxtbI7WVeUEozgR/g7rd0xUimYNlvZRE/K2MgZTjWy725lfeLeVcEM97mmtRGXw==",
"dev": true, "license": "MIT", "dependencies": { "binary-extensions": "^2.0.0" }, "engines": { "node": ">=8" } },
"node_modules/is-core-module": { "version": "2.16.1", "resolved": "https://registry.npmjs.org/is-core-module/-/is-
core-module-2.16.1.tgz", "integrity": "sha512-
UfoeMA6fJl8wTYFEUjelaGI67v6+N7qXJEvQuIGa99l4xsCruSYOVSSQ0uPANn4dAzm8lkYPaKLrrijLq7x23w==",
"dev": true, "license": "MIT", "dependencies": { "hasown": "^2.0.2" }, "engines": { "node": ">= 0.4" }, "funding": {
"url": "https://github.com/sponsors/ljharb" } }, "node_modules/is-extglob": { "version": "2.1.1", "resolved":
"https://registry.npmjs.org/is-extglob/-/is-extglob-2.1.1.tgz", "integrity": "sha512-
SbKbANKn603Vi4jEZv49LeVJMn4yGwsbzZworEoyEiutsN3nJYdbO36zfHGJ6QEDpOZIFkDtnq5JRxmvl3jsoQ==",
"dev": true, "license": "MIT", "engines": { "node": ">=0.10.0" } }, "node_modules/is-fullwidth-code-point": {
"version": "3.0.0", "resolved": "https://registry.npmjs.org/is-fullwidth-code-point/-/is-fullwidth-code-point-
3.0.0.tgz", "integrity": "sha512-
zymm5+u+sCsSWyD9qNaejV3DFVhCKclKdizYaJUuHA83RLjb7nSuGnddCHGv0hk+KY7BMAlsWeK4Ueg6EV6XQg
==", "dev": true, "license": "MIT", "engines": { "node": ">=8" } }, "node_modules/is-glob": { "version": "4.0.3",
"resolved": "https://registry.npmjs.org/is-glob/-/is-glob-4.0.3.tgz", "integrity": "sha512-
xelSayHH36ZgE7ZWWhli7pW34hNbNI8Ojv5KVmkJD4hBdD3th8Tfk9vYasLM+mXWOZhFkgZfxhLSnrwRr4elSSg==",
"dev": true, "license": "MIT", "dependencies": { "is-extglob": "^2.1.1" }, "engines": { "node": ">=0.10.0" } },
"node_modules/is-number": { "version": "7.0.0", "resolved": "https://registry.npmjs.org/is-number/-/is-number-

```


7.0.0.tgz", "integrity": "sha512-41Cifkg6e8TylSpdtTpeLVMqvSBEVzTttHvERD741+pnZ8ANv0004MRL43QKPDIK9cGvNp6NZWZUBlbGXYxxng==", "dev": true, "license": "MIT", "engines": { "node": ">=0.12.0" }, "node_modules/isexe": { "version": "2.0.0", "resolved": "https://registry.npmjs.org/isexe/-/isexe-2.0.0.tgz", "integrity": "sha512-RHxMLp9lnKHGHRng9QFhRCMbYAcVpn69smSGcq3f36xjgVvWThj4qqLbTLlq7Ssj8B+flQ1EuCEGI2IKsyQelw==", "dev": true, "license": "ISC", "node_modules/jackspeak": { "version": "3.4.3", "resolved": "https://registry.npmjs.org/jackspeak/-/jackspeak-3.4.3.tgz", "integrity": "sha512-OGlZQpz2yfahA/Rd1Y8Cd9SIEsqvXkLVoSv/cgwhnhFMDbsQFeZY0JJ7bIZBS9BcamUW96asq/npPWugM+RQBw==", "dev": true, "license": "BlueOak-1.0.0", "dependencies": { "@isaacs/cliui": "^8.0.2", "funding": { "url": "https://github.com/sponsors/isaacs" }, "optionalDependencies": { "@pkgjs/parseargs": "^0.11.0" }, "node_modules/jiti": { "version": "1.21.7", "resolved": "https://registry.npmjs.org/jiti/-/jiti-1.21.7.tgz", "integrity": "sha512-imKNG4EbWNRvJ0NC1H5/9GFy+taqGBHCaSSn+P2RnPgjsLmv6UD3Ej+Kj8nBWAraWyk7kK5ZUc+OEatnTR3A==", "dev": true, "license": "MIT", "bin": { "jiti": "bin/jiti.js" }, "node_modules/js-tokens": { "version": "4.0.0", "resolved": "https://registry.npmjs.org/js-tokens/-/js-tokens-4.0.0.tgz", "integrity": "sha512-RdJUflcE3cUzKiMqQgsCu06FPu9UdIJO0beYbPhHN4k6apGJtifoCtT9bcxOpYBtpD2kCM6Sbzg4CausW/PKQ==", "dev": true, "license": "MIT", "node_modules/js-yaml": { "version": "4.1.0", "resolved": "https://registry.npmjs.org/js-yaml/-/js-yaml-4.1.0.tgz", "integrity": "sha512-wpxZs9NoxZaJESJGIZTyDEaYpI0FKSA+FB9aJiyemKhMwKxQg63h4T1KJgUGHpTqPDNRcmmYLugrRjJBtWvRA==", "dev": true, "license": "MIT", "dependencies": { "argparse": "^2.0.1", "bin": { "js-yaml": "bin/js-yaml.js" }, "node_modules/jsesc": { "version": "3.1.0", "resolved": "https://registry.npmjs.org/jsesc/-/jsesc-3.1.0.tgz", "integrity": "sha512-/sM3dO2FOzXjKQhJuo0Q173wf2KOo8t4I8vHy6lF9poUp7bKT0/NHE8fPX23PwfhnkyfqnC2xRxOnVw5XuGlaA==", "dev": true, "license": "MIT", "bin": { "jsesc": "bin/jsesc" }, "engines": { "node": ">=6" }, "node_modules/json-buffer": { "version": "3.0.1", "resolved": "https://registry.npmjs.org/json-buffer/-/json-buffer-3.0.1.tgz", "integrity": "sha512-4bBV5BfR2mqfQTJm+V5tPPdf+ZpuhivTuAB5g8kcrXOZpTT/QwwVRWBywX1ozr6lEuPdbHxwaJlm9G6ml2sfSQ==", "dev": true, "license": "MIT", "node_modules/json-schema-traverse": { "version": "0.4.1", "resolved": "https://registry.npmjs.org/json-schema-traverse/-/json-schema-traverse-0.4.1.tgz", "integrity": "sha512-xbbCH5dCYU5T8LcEhhuh7HJ88HXuW3qsI3Y0zOZFKfZEHcpWiHU/Jxzk629Brsab/mMiHQti9wMP+845RPe3Vg==", "dev": true, "license": "MIT", "node_modules/json-stable-stringify-without-jsonify": { "version": "1.0.1", "resolved": "https://registry.npmjs.org/json-stable-stringify-without-jsonify/-/json-stable-stringify-without-jsonify-1.0.1.tgz", "integrity": "sha512-Bdboy+l7tA3OGW6FjyFHWkP5LuByj1Tk33Ljyq0axyzdk9/JSi2u3fP1QSmd1KNwq6VOKYGIaU87CisVir6Pw==", "dev": true, "license": "MIT", "node_modules/json5": { "version": "2.2.3", "resolved": "https://registry.npmjs.org/json5/-/json5-2.2.3.tgz", "integrity": "sha512-XmOWe7eyHYH14cLdVPoyg+GOH3rYX++KpzrylJwSW98tNk+U8XOI8FWKOGwtzdb8lXGf6zYwDUzeHMWfxayg==", "dev": true, "license": "MIT", "bin": { "json5": "lib/cli.js" }, "engines": { "node": ">=6" }, "node_modules/keyv": { "version": "4.5.4", "resolved": "https://registry.npmjs.org/keyv/-/keyv-4.5.4.tgz", "integrity": "sha512-V7Bp4ZyYDzqCYboq4ZUo07MyazGeuGxwq3hFpYoJv9ArI7sWbYzBMY4m6kUdQdovL5971dMuRk+13OI1Tg==", "dev": true, "license": "MIT", "dependencies": { "json-buffer": "3.0.1" }, "node_modules/levn": { "version": "0.4.1", "resolved": "https://registry.npmjs.org/levn/-/levn-0.4.1.tgz", "integrity": "sha512-+bT2GvwUa2RMseS+SrTBypEiwFqW7kwUpp8BwTMD8F85r6eF2ZSjYhJ0jZz8t+VFE/oYxeomVvduYpX4ppA==", "dev": true, "license": "MIT", "dependencies": { "prelude-ls": "^1.2.1", "type-check": "~0.4.0", "engines": { "node": ">= 0.8.0" }, "node_modules/lilconfig": { "version": "3.1.3", "resolved": "https://registry.npmjs.org/lilconfig/-/lilconfig-3.1.3.tgz", "integrity": "sha512-lqZ1B6/I9n+JaOYnyD/7b4EIkRiKz4GdnXP4OHLcT5JjjaymwtvK7F1gtdh8u8Wlo4CwG1HOdw8Rat6B5Dsg==", "dev": true, "license": "MIT", "engines": { "node": ">=14" }, "funding": { "url": "https://github.com/sponsors/antonk52" }, "node_modules/lines-and-columns": { "version": "1.2.4", "resolved": "https://registry.npmjs.org/lines-and-columns/-/lines-and-columns-1.2.4.tgz", "integrity": "sha512-169LqjSg3LQmVHPxtw8kLh3MQkKqSsT0hB4uWZqUdZiWtjczFqCYgdr0PYSzZvVp5sJ6/V/KJj8+1lFpyg=="

"https://registry.npmjs.org/lines-and-columns/-/lines-and-columns-1.2.4.tgz", "integrity": "sha512-7ylylesZQ/PV29jhEDI3Ufjo6ZX7gCqJr5F7PKrqc93v7fzSymt1BpwEU8nAUXs8qzzvqhbhK5QZg6Mt/HkBg==", "dev": true, "license": "MIT", "node_modules/locate-path": { "version": "6.0.0", "resolved": "https://registry.npmjs.org/locate-path/-/locate-path-6.0.0.tgz", "integrity": "sha512-iPZK6eYjbxRu3uB4/WZ3EsEIMJFMqAoopl3R+zuq0UjcAm/MO6KCweDgPfP3eITztoKP3KtnVHxTn2NHBSDVUw==", "dev": true, "license": "MIT", "dependencies": { "p-locate": "^5.0.0", "engines": { "node": ">=10" }, "funding": { "url": "https://github.com/sponsors/sindresorhus" }, "node_modules/lodash.merge": { "version": "4.6.2", "resolved": "https://registry.npmjs.org/lodash.merge/-/lodash.merge-4.6.2.tgz", "integrity": "sha512-0KpqXRvYyCsX1swR/XTK0va6VQkQM6MNO7PqW77ByjAhoARA8EfrP1N4+KIKj8YS0ZUCtRT/YUuhyYDujiQ==", "dev": true, "license": "MIT", "node_modules/lru-cache": { "version": "5.1.1", "resolved": "https://registry.npmjs.org/lru-cache/-/lru-cache-5.1.1.tgz", "integrity": "sha512-KpNARQA3lww+jTA0utUVVbrh+Jlrr1Fv0e56GGzAFOXN7dk/FviaDW8LHmK52DlC4HWP2n6gl8vN1aesBFgo9w==", "dev": true, "license": "ISC", "dependencies": { "yallist": "^3.0.2" }, "node_modules/lucide-react": { "version": "0.474.0", "resolved": "https://registry.npmjs.org/lucide-react/-/lucide-react-0.474.0.tgz", "integrity": "sha512-CmghgHkh0OJNmXGKwC0qfPJCYHASPMVSyGY8fj3xgk4v84ltqDg64JNKfZn5hC6E0vHi6gxnbcgwhyVB09wQtA==", "license": "ISC", "peerDependencies": { "react": "^16.5.1 || ^17.0.0 || ^18.0.0 || ^19.0.0" }, "node_modules/merge2": { "version": "1.4.1", "resolved": "https://registry.npmjs.org/merge2/-/merge2-1.4.1.tgz", "integrity": "sha512-8q7VEgMJW4J8tcfVPy8g09NcQwZdbwFEqhe/WZkolzn/3TGDwtOCYtXGxA3O8tPzpczCCDgv+P2P5y00ZJOog==", "dev": true, "license": "MIT", "engines": { "node": ">= 8" }, "node_modules/micromatch": { "version": "4.0.8", "resolved": "https://registry.npmjs.org/micromatch/-/micromatch-4.0.8.tgz", "integrity": "sha512-PXwfBhYU0hBCPw8Dn0E+WDYb7af3dSLVWKi3HGv84ldF4TyFoC0ysxFd0Goxw7nSv4T/PzEJQxsYsEiFCKo2BA==", "dev": true, "license": "MIT", "dependencies": { "braces": "^3.0.3", "picomatch": "^2.3.1", "engines": { "node": ">=8.6" }, "node_modules/micromatch/node_modules/picomatch": { "version": "2.3.1", "resolved": "https://registry.npmjs.org/picomatch/-/picomatch-2.3.1.tgz", "integrity": "sha512-JU3teHTNjmE2VCGFzuY8EXzCDVwEqB2a8fslwaStHhAWJEEvd1o1QD80CU6+ZdEXXSLbSsuLwJjkCBWqRQUVA==", "dev": true, "license": "MIT", "engines": { "node": ">=8.6" }, "funding": { "url": "https://github.com/sponsors/jonschlinkert" }, "node_modules/minimatch": { "version": "3.1.2", "resolved": "https://registry.npmjs.org/minimatch/-/minimatch-3.1.2.tgz", "integrity": "sha512-J7p63hRiAjw1NDEww1W7i37+BylrOWO5XQQAzZ3VOcL0PNybwpmfV/N05zFAzwQ9USyEcX6t3UO+K5aqBQOIHw==", "dev": true, "license": "ISC", "dependencies": { "brace-expansion": "^1.1.7", "engines": { "node": " " }, "node_modules/minipass": { "version": "7.1.2", "resolved": "https://registry.npmjs.org/minipass/-/minipass-7.1.2.tgz", "integrity": "sha512-qOOzS1cBTWYF4BH8fVePDBOO9iptMnGUEZwNc/cMWnTV2nVLZ7VoNWEPhkYczZA0pdoA7dl6e7FL659nX9S2aw==", "dev": true, "license": "ISC", "engines": { "node": ">=16 || 14 >=14.17" }, "node_modules/ms": { "version": "2.1.3", "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz", "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhizq5oxgHyaCU9yYXvcLsvoVaHJq/s5xI6/XXP6tz7R9xAOtHnSO/tXtF3WRTIA==", "dev": true, "license": "MIT", "node_modules/mz": { "version": "2.7.0", "resolved": "https://registry.npmjs.org/mz/-/mz-2.7.0.tgz", "integrity": "sha512-z81GNO7nnYMEhrGh9LeymoE4+Yr0Wn5McHIZMK5cfQCI+NDX08sCZgUc9/6MHni9WuFLm1Z3HTCXu2z9fN62Q==", "dev": true, "license": "MIT", "dependencies": { "any-promise": "^1.0.0", "object-assign": "^4.0.1", "thenify-all": "^1.0.0" }, "node_modules/nanoid": { "version": "3.3.11", "resolved": "https://registry.npmjs.org/nanoid/-/nanoid-3.3.11.tgz", "integrity": "sha512-N8SpfPUnUp1bK+PMYW8qSWdl9U+wwNWI4QKxOYDy9JAro3WMX7p2OeVRF9v+347pnakNevPmiHhNmZ2HbFA76w==", "dev": true, "funding": [{ "type": "github", "url": "https://github.com/sponsors/ai" }], "license": "MIT", "bin": { "nanoid": "bin/nanoid.cjs" }, "engines": { "node": ">=10 || ^12 || ^13.7 || ^14 || >=15.0.1" }, "node_modules/natural-compare": { "version": "1.4.0", "resolved": "https://registry.npmjs.org/natural-compare/-/natural-compare-1.4.0.tgz", "integrity": "sha512-OWND8ei3VtNC9h7V60qff3SVobHr996CTwgxubgyQYEpg290h9J0buyECNNJexkFm5sOajh5G116RYA1c8ZMSw=

```
=, "dev": true, "license": "MIT" }, "node_modules/node-releases": { "version": "2.0.21", "resolved":  
"https://registry.npmjs.org/node-releases/-/node-releases-2.0.21.tgz", "integrity": "sha512-  
5b0pgg78U3hwXkCM8Z9b2FJdPZlr9Psr9V2gQPESdGHqbntYFJKFW4r5TeWGFzafGY3hzs1JC62VEQMbl1JFkw==  
", "dev": true, "license": "MIT" }, "node_modules/normalize-path": { "version": "3.0.0", "resolved":  
"https://registry.npmjs.org/normalize-path/-/normalize-path-3.0.0.tgz", "integrity": "sha512-  
6eZs5Ls3WtCisHWp9S2GUy8dqkpGi4BVSz3GaqiE6ezub0512ESztXUwUB6C6IKbQkY2Pnb/mD4WYojCRwcwLA=  
=", "dev": true, "license": "MIT", "engines": { "node": ">=0.10.0" } }, "node_modules/normalize-range": { "version":  
"0.1.2", "resolved": "https://registry.npmjs.org/normalize-range/-/normalize-range-0.1.2.tgz", "integrity": "sha512-  
bdok/XvKil3nUpKlnV6P2hxtMNRcBoOjAcyBuQnWEhO665FwrSNRrXU+AqpsvO6LgGYPspN+lu5CLtw4jPRKNA==",  
"dev": true, "license": "MIT", "engines": { "node": ">=0.10.0" } }, "node_modules/object-assign": { "version":  
"4.1.1", "resolved": "https://registry.npmjs.org/object-assign/-/object-assign-4.1.1.tgz", "integrity": "sha512-  
rJgTQnkUnH1sFw8yT6VSU3zD3sWmu6sZhIseY8VX+GRu3P6F7Fu+JND0XfklElbLJSnc3FUQHVe4cU5hj+BcUg==  
", "dev": true, "license": "MIT", "engines": { "node": ">=0.10.0" } }, "node_modules/object-hash": { "version":  
"3.0.0", "resolved": "https://registry.npmjs.org/object-hash/-/object-hash-3.0.0.tgz", "integrity": "sha512-  
RSn9F68PjH9HqtltsSnqYC1XXoWe9Bju5+213R98cNGttag9q9yAOTzdbsqvIa7aNm5WffBZFpWYr2aWrklWAw==",  
"dev": true, "license": "MIT", "engines": { "node": ">= 6" } }, "node_modules/optionator": { "version": "0.9.4",  
"resolved": "https://registry.npmjs.org/optionator/-/optionator-0.9.4.tgz", "integrity": "sha512-  
6lpQ7mKUxRcZNLIObR0hz7lxsapSSiYNZJwXPGeF0mTVqGKfIXj1DQcMoT22S3ROcLyY/rz0PWaWZ9ayWmad9g=  
=", "dev": true, "license": "MIT", "dependencies": { "deep-is": "^0.1.3", "fast-levenshtein": "^2.0.6", "levn":  
"^0.4.1", "prelude-ls": "^1.2.1", "type-check": "^0.4.0", "word-wrap": "^1.2.5" }, "engines": { "node": ">= 0.8.0" } },  
"node_modules/p-limit": { "version": "3.1.0", "resolved": "https://registry.npmjs.org/p-limit/-/p-limit-3.1.0.tgz",  
"integrity": "sha512-  
TYOanM3wGwNGsZN2cVTYPArw454xnXj5qmWF1bEoAc4+cU/ol7GVh7odevjp1FNHduHc3KZMcFduxU5Xc6uJRQ  
==", "dev": true, "license": "MIT", "dependencies": { "yocto-queue": "^0.1.0" }, "engines": { "node": ">=10" },  
"funding": { "url": "https://github.com/sponsors/sindresorhus" } }, "node_modules/p-locate": { "version": "5.0.0",  
"resolved": "https://registry.npmjs.org/p-locate/-/p-locate-5.0.0.tgz", "integrity": "sha512-  
LaNjtRWUBY++zB5nE/NwcaoMylSPk+S+ZHNb1TzdbMJMny6dynpAGt7X/tl/QYq3TleE6nxHppbo2LGymrG5Pw==",  
"dev": true, "license": "MIT", "dependencies": { "p-limit": "^3.0.2" }, "engines": { "node": ">=10" }, "funding": {  
"url": "https://github.com/sponsors/sindresorhus" } }, "node_modules/package-json-from-dist": { "version":  
"1.0.1", "resolved": "https://registry.npmjs.org/package-json-from-dist/-/package-json-from-dist-1.0.1.tgz",  
"integrity": "sha512-  
UEZIS3/by4OC8vL3P2dTXRETpebLI2Nil5vIrjaD/5UtrkFX/tNbwjTSRAGC/+7CAo2plcBaRgWmcBBHcsaClw==",  
"dev": true, "license": "BlueOak-1.0.0" }, "node_modules/parent-module": { "version": "1.0.1", "resolved":  
"https://registry.npmjs.org/parent-module/-/parent-module-1.0.1.tgz", "integrity": "sha512-  
GQ2EWRpQV8/o+Aw8YqtfZZPfNRWZYkbidE9k5rpl/hC3vtHhBfGm2lfi6qVW+coDGkrUKZAx3Lot5kcsRIh+g==",  
"dev": true, "license": "MIT", "dependencies": { "callsites": "^3.0.0" }, "engines": { "node": ">=6" } },  
"node_modules/path-exists": { "version": "4.0.0", "resolved": "https://registry.npmjs.org/path-exists/-/path-exists-  
4.0.0.tgz", "integrity": "sha512-  
ak9Qy5Q7jYb2Wwcey5Fpvg2KoAc/ZiHLSLOSBmRmygPsGwkVVt0fZa0qrtMz+m6tJTAHfZQ8FnmB4MG4LWy7/w=  
=", "dev": true, "license": "MIT", "engines": { "node": ">=8" } }, "node_modules/path-key": { "version": "3.1.1",  
"resolved": "https://registry.npmjs.org/path-key/-/path-key-3.1.1.tgz", "integrity": "sha512-  
ojmeN0qd+y0jszEtoY48r0Peq5dwMEkIICOu6Q5f41lfksWxKuTyrhgoTpLnlycHm24Uhqx+5Tqm2lnSwLhE6Q==",  
"dev": true, "license": "MIT", "engines": { "node": ">=8" } }, "node_modules/path-parse": { "version": "1.0.7",  
"resolved": "https://registry.npmjs.org/path-parse/-/path-parse-1.0.7.tgz", "integrity": "sha512-  
LDJzPVEEEPR+y48z93A0Ed0yXb8pABYGW0/k5YYdYgpY2/2EsOsksJrq7IOHxryrVOn1ejG6oAp8ahvOIQD8sw==",  
"dev": true, "license": "MIT" }, "node_modules/path-scurry": { "version": "1.11.1", "resolved":  
"https://registry.npmjs.org/path-scurry/-/path-scurry-1.11.1.tgz", "integrity": "sha512-  
Xa4Nw17FS9ApQFJ9umLiJS4orGjm7ZzwUrwmamGQuHSzDyth9boKDaycYdDcZDuqYATXw4HFXgaqWTctW/v1HA  
==", "dev": true, "license": "BlueOak-1.0.0", "dependencies": { "lru-cache": "^10.2.0", "minipass": "^5.0.0 || ^6.0.2
```

```

|| ^7.0.0" }, "engines": { "node": ">=16 || 14 >=14.18" }, "funding": { "url": "https://github.com/sponsors/isaacs" } },
"node_modules/path-scurry/node_modules/lru-cache": { "version": "10.4.3", "resolved":
"https://registry.npmjs.org/lru-cache/-/lru-cache-10.4.3.tgz", "integrity": "sha512-
JNAzZcXrCt42VGLuYz0zfAzDfAvJWW6AfYIDBQyDV5DCII2m5sAmK+OIO7s59XfsRsWHp02jAJrRadPRGTt6SQ=="
, "dev": true, "license": "ISC" }, "node_modules/picocolors": { "version": "1.1.1", "resolved":
"https://registry.npmjs.org/picocolors/-/picocolors-1.1.1.tgz", "integrity": "sha512-
xceH2snhtb5M9liqDsmEw56le376mTZkEX/jEb/RxNFyegNul7eNslCXP9FDj/Lcu0X8KEyMceP2ntpaHrDEVA==",
"dev": true, "license": "ISC" }, "node_modules/picomatch": { "version": "4.0.3", "resolved":
"https://registry.npmjs.org/picomatch/-/picomatch-4.0.3.tgz", "integrity": "sha512-
5gTmgEY/sqK6gFXLIsQNH19IWb4ebPDLA4SdLP7dsWkIXHWIG66oPuVvXSGFPppYZz8ZDZq0dYYrbHfBCVUb1Q
==", "dev": true, "license": "MIT", "engines": { "node": ">=12" }, "funding": { "url":
"https://github.com/sponsors/jonschlinkert" } }, "node_modules/pify": { "version": "2.3.0", "resolved":
"https://registry.npmjs.org/pify/-/pify-2.3.0.tgz", "integrity": "sha512-
udgsAY+fTnvv7kl7aaxbqwWNb0AHiB0qBO89PZKPkoTmGOgdbHDKD+0B2X4uTfJ/FT1R09r9gTsjUjNjotuog==",
"dev": true, "license": "MIT", "engines": { "node": ">=0.10.0" } }, "node_modules/pirates": { "version": "4.0.7",
"resolved": "https://registry.npmjs.org/pirates/-/pirates-4.0.7.tgz", "integrity": "sha512-
TfySrs/5nm8fQJDcBDuUng3VOUKsd7S+qzvbOTiGXHfxX4wK31ard+hoNuvkicM/2YFzlpDgABOevKSsB4G/FA==",
"dev": true, "license": "MIT", "engines": { "node": ">= 6" } }, "node_modules/postcss": { "version": "8.5.6",
"resolved": "https://registry.npmjs.org/postcss/-/postcss-8.5.6.tgz", "integrity": "sha512-
3Ybi1tAuWAP9s0r1UQ2J4n5Y0G05bJkpUIO0/bl9MhwmD70S5aTWbXGBwxHrelT+XM1k6dM0pk+SwNkpTRN7Pg=
=", "dev": true, "funding": [ { "type": "opencollective", "url": "https://opencollective.com/postcss/" }, { "type":
"tidelift", "url": "https://tidelift.com/funding/github/npm/postcss" }, { "type": "github", "url":
"https://github.com/sponsors/ai" } ], "license": "MIT", "dependencies": { "nanoid": "^3.3.11", "picocolors": "^1.1.1",
"source-map-js": "^1.2.1" }, "engines": { "node": "^10 || ^12 || >=14" } }, "node_modules/postcss-import": {
"version": "15.1.0", "resolved": "https://registry.npmjs.org/postcss-import/-/postcss-import-15.1.0.tgz", "integrity":
"sha512-
hpr+J05B2FVYUAXHeK1Yyl267J/dDDhMU6B6civm8hSY1jYJnBXxzKDKDswzJmtLHryrjhnDjqqp/49t8FALew==",
"dev": true, "license": "MIT", "dependencies": { "postcss-value-parser": "^4.0.0", "read-cache": "^1.0.0",
"resolve": "^1.1.7" }, "engines": { "node": ">=14.0.0" }, "peerDependencies": { "postcss": "^8.0.0" },
"node_modules/postcss-js": { "version": "4.1.0", "resolved": "https://registry.npmjs.org/postcss-js/-/postcss-js-
4.1.0.tgz", "integrity": "sha512-
oIAOTqglo7q2EOwbhb8UaIYePMvYoleRY2YKntdpFQXNosSu3vLrniGgmH9OKs/qAkfoj5oB3le/7mINW1LCfw==",
"dev": true, "funding": [ { "type": "opencollective", "url": "https://opencollective.com/postcss/" }, { "type":
"github", "url": "https://github.com/sponsors/ai" } ], "license": "MIT", "dependencies": { "camelcase-css": "^2.0.1"
}, "engines": { "node": "^12 || ^14 || >= 16" }, "peerDependencies": { "postcss": "^8.4.21" } },
"node_modules/postcss-load-config": { "version": "6.0.1", "resolved": "https://registry.npmjs.org/postcss-load-
config/-/postcss-load-config-6.0.1.tgz", "integrity": "sha512-
oPtTM4oerL+UXmx+93ytZVN82RrY/wPUV8leDxFrZljXOLF1pN+EmKPLbubvKHT2HC20xXsCAH2Z+CKV6Oz/g==",
"dev": true, "funding": [ { "type": "opencollective", "url": "https://opencollective.com/postcss/" }, { "type":
"github", "url": "https://github.com/sponsors/ai" } ], "license": "MIT", "dependencies": { "lilconfig": "^3.1.1" },
"engines": { "node": ">= 18" }, "peerDependencies": { "jiti": ">=1.21.0", "postcss": ">=8.0.9", "tsx": "^4.8.1",
"yaml": "^2.4.2" }, "peerDependenciesMeta": { "jiti": { "optional": true }, "postcss": { "optional": true }, "tsx": {
"optional": true }, "yaml": { "optional": true } } }, "node_modules/postcss-nested": { "version": "6.2.0", "resolved":
"https://registry.npmjs.org/postcss-nested/-/postcss-nested-6.2.0.tgz", "integrity": "sha512-
HQBt28KulC5AHzG+cZtj9kvKB93CFCdLvog1WFLf1D+xmMvPGIBstkpTEZfK5+AN9hfJocyBFCNiqyS48bpgzQ==",
"dev": true, "funding": [ { "type": "opencollective", "url": "https://opencollective.com/postcss/" }, { "type":
"github", "url": "https://github.com/sponsors/ai" } ], "license": "MIT", "dependencies": { "postcss-selector-parser":
"^6.1.1" }, "engines": { "node": ">=12.0" }, "peerDependencies": { "postcss": "^8.2.14" } },
"node_modules/postcss-selector-parser": { "version": "6.1.2", "resolved": "https://registry.npmjs.org/postcss-

```

```

selector-parser/-/postcss-selector-parser-6.1.2.tgz", "integrity": "sha512-
Q8qQfPiZ+THO/3ZrOrO0cJJkfpYCagtMUKXbnEfmGujwXg6z/WBeOyS9APBBPCTSiDV+s4SwQGu8yFsiMRludg==
", "dev": true, "license": "MIT", "dependencies": { "cssesc": "^3.0.0", "util-deprecate": "^1.0.2" }, "engines": {
"node": ">=4" } }, "node_modules/postcss-value-parser": { "version": "4.2.0", "resolved":
"https://registry.npmjs.org/postcss-value-parser/-/postcss-value-parser-4.2.0.tgz", "integrity": "sha512-
1NNCs6uurfkVbeXG4S8JFT9t19m45lCnif8zWLd5oPSZ50QnwMfK+H3jv408d4jw/7Bttv5axS5liHoLaVNHeQ==",
"dev": true, "license": "MIT" }, "node_modules/prelude-ls": { "version": "1.2.1", "resolved":
"https://registry.npmjs.org/prelude-ls/-/prelude-ls-1.2.1.tgz", "integrity": "sha512-
vkcDPrRZo1QZLbn5RLGPpg/WmlQ65qoWWhcGKf/b5epIkKX0m9z8ppCat4mlOqUsWpyNuYgO3VRyrYHSzX5g=
=", "dev": true, "license": "MIT", "engines": { "node": ">= 0.8.0" } }, "node_modules/punycode": { "version":
"2.3.1", "resolved": "https://registry.npmjs.org/punycode/-/punycode-2.3.1.tgz", "integrity": "sha512-
vYt7UD1U9Wg6138shLtLOvdAu+8DsC/ilFtEVHcH+wYdcSpNE20AfSOduf6MkRFahL5FY7X1oU7nKVZFtfq8Fg==",
"dev": true, "license": "MIT", "engines": { "node": ">=6" } }, "node_modules/queue-microtask": { "version":
"1.2.3", "resolved": "https://registry.npmjs.org/queue-microtask/-/queue-microtask-1.2.3.tgz", "integrity": "sha512-
NuaNSa6fIKT5JaSYQzJok04JzTL1CA6aGhv5rLW3PgqA+M2ChpZQnAC8h8i4ZFkBS8X5RqkDBHA7r4hej3K9A==",
"dev": true, "funding": [ { "type": "github", "url": "https://github.com/sponsors/feross" }, { "type": "patreon", "url":
"https://www.patreon.com/feross" }, { "type": "consulting", "url": "https://feross.org/support" } ], "license": "MIT"
}, "node_modules/react": { "version": "19.1.1", "resolved": "https://registry.npmjs.org/react/-/react-19.1.1.tgz",
"integrity": "sha512-
w8nqGlmo45dmMlfljMwOGtbmC/mk4CMYhWlicdSflH91J9TyCyczcPFXJzrZ/ZXcgGRFeP6BU0BEJTw6tZdfQ==",
"license": "MIT", "engines": { "node": ">=0.10.0" } }, "node_modules/react-dom": { "version": "19.1.1", "resolved":
"https://registry.npmjs.org/react-dom/-/react-dom-19.1.1.tgz", "integrity": "sha512-
Dlq/5LAZgF0Gaz6yiqZCf6VCcZs1ghAJyrsu84Q/GT0gV+mCxbfmKNOGRKBYMJ8IEdGPQu49YWXD02GCKnEDkw=
=", "license": "MIT", "dependencies": { "scheduler": "^0.26.0" }, "peerDependencies": { "react": "^19.1.1" },
"node_modules/react-refresh": { "version": "0.17.0", "resolved": "https://registry.npmjs.org/react-refresh/-/react-
refresh-0.17.0.tgz", "integrity": "sha512-
z6F7K9bV85EfseRCp2bzrpyQ0Gkw1uLoCel9XBVWPg/TjRj94SkJzUTGfOa4bs7iJvBWtQG0Wq7wnl0syw3EBQ==",
"dev": true, "license": "MIT", "engines": { "node": ">=0.10.0" } }, "node_modules/react-remove-scroll": { "version":
"2.7.1", "resolved": "https://registry.npmjs.org/react-remove-scroll/-/react-remove-scroll-2.7.1.tgz", "integrity":
"sha512-
HpMh8+oahmldOuS5aFKKY6Pyog+FNaZV/XyJOq7b4YFwsFHe5yYfdblll4k3vU2nSDql7YskmUseHsRrJqIPA==",
"license": "MIT", "dependencies": { "react-remove-scroll-bar": "^2.3.7", "react-style-singleton": "^2.2.3", "tslib":
"^2.1.0", "use-callback-ref": "^1.3.3", "use-sidecar": "^1.1.3" }, "engines": { "node": ">=10" },
"peerDependencies": { "@types/react": " ", "react": "^16.8.0 || ^17.0.0 || ^18.0.0 || ^19.0.0 || ^19.0.0-rc" },
"peerDependenciesMeta": { "@types/react": { "optional": true } } }, "node_modules/react-remove-scroll-bar": {
"version": "2.3.8", "resolved": "https://registry.npmjs.org/react-remove-scroll-bar/-/react-remove-scroll-bar-
2.3.8.tgz", "integrity": "sha512-
9r+yi9+mgU33AKcj6lbT9oRCO78WriSj6t/cF8DWBZJ9aOGPOTEDvdUDz1FwKim7QXWwmHqtdHnRjfhAxEG46Q
==", "license": "MIT", "dependencies": { "react-style-singleton": "^2.2.2", "tslib": "^2.0.0" }, "engines": { "node":
">=10" }, "peerDependencies": { "@types/react": " ", "react": "^16.8.0 || ^17.0.0 || ^18.0.0 || ^19.0.0" },
"peerDependenciesMeta": { "@types/react": { "optional": true } } }, "node_modules/react-router": { "version":
"6.30.1", "resolved": "https://registry.npmjs.org/react-router/-/react-router-6.30.1.tgz", "integrity": "sha512-
X1m21aEmxGXqENEPG3T6u0Th7g0aS4ZmoNynhbs+Cn+q+QGTLt+d5lQ2bHAXKzKcxGJxACPvbnYQSCRcfxHIQ
==", "dependencies": { "@remix-run/router": "1.23.0" }, "engines": { "node": ">=14.0.0" }, "peerDependencies": {
"react": ">=16.8" } }, "node_modules/react-router-dom": { "version": "6.30.1", "resolved":
"https://registry.npmjs.org/react-router-dom/-/react-router-dom-6.30.1.tgz", "integrity": "sha512-
lIKsgOkZdbPU1Eg3zK8lCn+sJD9wMRZZPuzmdWWX5SUs8OFkN5HnFVC0u5KMeMaC9aoancFI/KoLuKPqN+hxHw
==", "dependencies": { "@remix-run/router": "1.23.0", "react-router": "6.30.1" }, "engines": { "node": ">=14.0.0" },
"peerDependencies": { "react": ">=16.8", "react-dom": ">=16.8" } }, "node_modules/react-style-singleton": {

```

```
"version": "2.2.3", "resolved": "https://registry.npmjs.org/react-style-singleton/-/react-style-singleton-2.2.3.tgz",
"integrity": "sha512-
b6jSvxxVnyptAiLjkbWLE/IOhR4lftDAI+eUC7RZy+QQWc6wRzIV2CE6xBuMmDxc2qlihtDCZD5NPOFI7fRBQ==",
"license": "MIT", "dependencies": { "get-nonce": "^1.0.0", "tslib": "^2.0.0" }, "engines": { "node": ">=10" },
"peerDependencies": { "@types/react": " " }, "react": "^16.8.0 || ^17.0.0 || ^18.0.0 || ^19.0.0 || ^19.0.0-rc",
"peerDependenciesMeta": { "@types/react": { "optional": true } }, "node_modules/read-cache": { "version":
"1.0.0", "resolved": "https://registry.npmjs.org/read-cache/-/read-cache-1.0.0.tgz", "integrity": "sha512-
Owdv/Ft7lJlOgm/i0xvNDZ1LrRANRfew4b2prF3OWMQLxLfu3bS8FVhCsrSCMK4IR56Y9ya+AThOTpDCTxCmpRA==",
, "dev": true, "license": "MIT", "dependencies": { "pify": "^2.3.0" }, "node_modules/readdirp": { "version": "3.6.0",
"resolved": "https://registry.npmjs.org/readdirp/-/readdirp-3.6.0.tgz", "integrity": "sha512-
hOS089on8RduqdbbhvQ5Z37A0ESjsqz6qnRcfsMU3495FuTdqSm+7bhJ29JvIOsBDEEnan5DPu9t3To9VRIMzA==",
, "dev": true, "license": "MIT", "dependencies": { "picomatch": "^2.2.1", "engines": { "node": ">=8.10.0" } },
"node_modules/readdirp/node_modules/picomatch": { "version": "2.3.1", "resolved":
"https://registry.npmjs.org/picomatch/-/picomatch-2.3.1.tgz", "integrity": "sha512-
JU3teHTNjmE2VCGFzuY8EXzCDVwEqB2a8fslvwaStHhAWJEEvd1o1QD80CU6+ZdEXXSLbSsuLwJjkCBWqRQUVA
==", "dev": true, "license": "MIT", "engines": { "node": ">=8.6" }, "funding": { "url":
"https://github.com/sponsors/jonschlinkert" } }, "node_modules/resolve": { "version": "1.22.10", "resolved":
"https://registry.npmjs.org/resolve/-/resolve-1.22.10.tgz", "integrity": "sha512-
NPRy+ncIMeDITAsuqwkIfierawhefJtkNSW0qZJEqMEb+qBt/77B/jGeeek+F0uOeN05CDa6HXbbIgtVX4w==",
"dev": true, "license": "MIT", "dependencies": { "is-core-module": "^2.16.0", "path-parse": "^1.0.7", "supports-
preserve-symlinks-flag": "^1.0.0", "bin": { "resolve": "bin/resolve" }, "engines": { "node": ">= 0.4" }, "funding": {
"url": "https://github.com/sponsors/ljharb" } }, "node_modules/resolve-from": { "version": "4.0.0", "resolved":
"https://registry.npmjs.org/resolve-from/-/resolve-from-4.0.0.tgz", "integrity": "sha512-
pb/MYmXstAkysRFx8piN1tGFNQIFA3vkE3Gq4EuA1dF6gHp/+vgZqsCGJapvy8N3Q+4o7FwvquPJcnZ7RYy4g==",
"dev": true, "license": "MIT", "engines": { "node": ">=4" }, "node_modules/reusify": { "version": "1.1.0", "resolved":
"https://registry.npmjs.org/reusify/-/reusify-1.1.0.tgz", "integrity": "sha512-
g6QUff04oZpHs0eG5p83rFLhHeV00ug/Yf9nZM6fLeUrPguBTkTQOdpAWWspMh55TZfVQDPa3NQJfbVRAXdlw
==", "dev": true, "license": "MIT", "engines": { "iojs": ">=1.0.0", "node": ">=0.10.0" }, "node_modules/rollup": {
"version": "4.50.2", "resolved": "https://registry.npmjs.org/rollup/-/rollup-4.50.2.tgz", "integrity": "sha512-
BgLRGy7tNS9H66aIMASq1qSYbAAJV6Z6WR4QYTvj5FgF15rZ/ympt1uixHXwzbZUBDbkvqUI1KR0fH1FhMaQ9w=
=", "dev": true, "license": "MIT", "dependencies": { "@types/estree": "1.0.8", "bin": { "rollup": "dist/bin/rollup" },
"engines": { "node": ">=18.0.0", "npm": ">=8.0.0", "optionalDependencies": { "@rollup/rollup-android-arm-eabi":
"4.50.2", "@rollup/rollup-android-arm64": "4.50.2", "@rollup/rollup-darwin-arm64": "4.50.2", "@rollup/rollup-
darwin-x64": "4.50.2", "@rollup/rollup-freebsd-arm64": "4.50.2", "@rollup/rollup-freebsd-x64": "4.50.2",
"@rollup/rollup-linux-arm-gnueabi": "4.50.2", "@rollup/rollup-linux-arm-musleabi": "4.50.2", "@rollup/rollup-
linux-arm64-gnu": "4.50.2", "@rollup/rollup-linux-arm64-musl": "4.50.2", "@rollup/rollup-linux-loong64-gnu":
"4.50.2", "@rollup/rollup-linux-ppc64-gnu": "4.50.2", "@rollup/rollup-linux-riscv64-gnu": "4.50.2", "@rollup/rollup-
linux-riscv64-musl": "4.50.2", "@rollup/rollup-linux-s390x-gnu": "4.50.2", "@rollup/rollup-linux-x64-gnu":
"4.50.2", "@rollup/rollup-linux-x64-musl": "4.50.2", "@rollup/rollup-openharmony-arm64": "4.50.2",
"@rollup/rollup-win32-arm64-msvc": "4.50.2", "@rollup/rollup-win32-ia32-msvc": "4.50.2", "@rollup/rollup-
win32-x64-msvc": "4.50.2", "fsevents": "~2.3.2" } }, "node_modules/run-parallel": { "version": "1.2.0", "resolved":
"https://registry.npmjs.org/run-parallel/-/run-parallel-1.2.0.tgz", "integrity": "sha512-
5l4VyZR86LZ/IDxZTR6jqL8AFE2S0IFLMP26AbjsLVADxHdhB/c0GUsH+y39UfCi3dzz8OIQuPmnaJOMoDHQBA==",
, "dev": true, "funding": [ { "type": "github", "url": "https://github.com/sponsors/feross" }, { "type": "patreon", "url":
"https://www.patreon.com/feross" }, { "type": "consulting", "url": "https://feross.org/support" } ], "license": "MIT",
"dependencies": { "queue-microtask": "^1.2.2" }, "node_modules/scheduler": { "version": "0.26.0", "resolved":
"https://registry.npmjs.org/scheduler/-/scheduler-0.26.0.tgz", "integrity": "sha512-
NIHwtCI/5gCPR3D1nNXtWABUmBwwZpEQiD4IXSbiDq8BzLIK/7Ir5gTFSGZDUu37K5cMNp0hFtzO38sC7gWA==",
"license": "MIT" }, "node_modules/semver": { "version": "6.3.1", "resolved":
```

"https://registry.npmjs.org/semver/-/semver-6.3.1.tgz", "integrity": "sha512-
BR7VvDCVHO+q2xBEWskxS6DJE1qRnb7DxzUrogb71CWoSfciBxYsiAGd+KI0mmq/MprG9yArRkyrQxTO6XjMzA==
", "dev": true, "license": "ISC", "bin": { "semver": "bin/semver.js" }, "node_modules/shebang-command": {
"version": "2.0.0", "resolved": "https://registry.npmjs.org/shebang-command/-/shebang-command-2.0.0.tgz",
"integrity": "sha512-
kHxr2ZpYtdmrN1qDjrrX/Z1rR1kG8Dx+gkpK1G4eXmvXswmcE1hTWBWWYUzIraYw1/yZp6YuDY77YtvbN0dmDA==",
"dev": true, "license": "MIT", "dependencies": { "shebang-regex": "^3.0.0" }, "engines": { "node": ">=8" },
"node_modules/shebang-regex": { "version": "3.0.0", "resolved": "https://registry.npmjs.org/shebang-
regex/-/shebang-regex-3.0.0.tgz", "integrity": "sha512-
7++dFhtcx3353uBaq8DDR4NuxBetBzC7ZQOhmTQInHEd6bSrXdiEyzCvG07Z44UYdLShWUyXt5M/yhz8ekcb1A=
=", "dev": true, "license": "MIT", "engines": { "node": ">=8" }, "node_modules/signal-exit": { "version": "4.1.0",
"resolved": "https://registry.npmjs.org/signal-exit/-/signal-exit-4.1.0.tgz", "integrity": "sha512-
bzyZ1e88w9O1iNjBKnOlVYTrWPDl46O1bG0D3XInv+9tkPrxrN8jUUTiFIDkkmKWgn1M6CfIA13SuGqOa9Korw==",
"dev": true, "license": "ISC", "engines": { "node": ">=14" }, "funding": { "url": "https://github.com/sponsors/isaacs"
} }, "node_modules/source-map-js": { "version": "1.2.1", "resolved": "https://registry.npmjs.org/source-map-
js/-/source-map-js-1.2.1.tgz", "integrity": "sha512-
UXWMKhLOWVKb728lUtQPXxfYU+usdybtUrK/8uGE8CQMvrhOpwvzDBwj0QhSL7MQc7vIsISBG8VQ8+IDQxpFQA
==", "dev": true, "license": "BSD-3-Clause", "engines": { "node": ">=0.10.0" }, "node_modules/string-width": {
"version": "5.1.2", "resolved": "https://registry.npmjs.org/string-width/-/string-width-5.1.2.tgz", "integrity":
"sha512-
HnLOCR3vjY8beoNLtcjZ5/nxn2afmME6lhrDrebokqMap+XbeW8n9TXpPDOqdGK5qcI3oT0GKTW6wC7EMiVqA==
", "dev": true, "license": "MIT", "dependencies": { "eastasianwidth": "^0.2.0", "emoji-regex": "^9.2.2", "strip-ansi":
"7.0.1" }, "engines": { "node": ">=12" }, "funding": { "url": "https://github.com/sponsors/sindresorhus" },
"node_modules/string-width-cjs": { "name": "string-width", "version": "4.2.3", "resolved":
"https://registry.npmjs.org/string-width/-/string-width-4.2.3.tgz", "integrity": "sha512-
wKyQRQpjJ0slIp62ErSZdGsjMJWsap50RNihHhu6G7JVO/9jIB6UyevL+tXuOqrng8j/cxKTWYUWwvSTriiZz/g==",
"dev": true, "license": "MIT", "dependencies": { "emoji-regex": "^8.0.0", "is-fullwidth-code-point": "^3.0.0", "strip-
ansi": "^6.0.1" }, "engines": { "node": ">=8" }, "node_modules/string-width-cjs/node_modules/ansi-regex": {
"version": "5.0.1", "resolved": "https://registry.npmjs.org/ansi-regex/-/ansi-regex-5.0.1.tgz", "integrity": "sha512-
quJQXITSUGL2LH9SUXo8VwsY4soanhgo6LNSm84E1LBcE8s3O0wpdiRzyR9z/ZZJMIMWv37qOOb9pdJIMUEKFQ
==", "dev": true, "license": "MIT", "engines": { "node": ">=8" }, "node_modules/string-width-
cjs/node_modules/emoji-regex": { "version": "8.0.0", "resolved": "https://registry.npmjs.org/emoji-regex/-/emoji-
regex-8.0.0.tgz", "integrity": "sha512-
MSjYzcWNOA0ewAHpz0MxpYFvwg6yjj1NG3xteoqz644VCo/RPgnr1/GGt+ic3iJTzQ8Eu3TdM14SawnVUmGE6A==
", "dev": true, "license": "MIT" }, "node_modules/string-width-cjs/node_modules/strip-ansi": { "version": "6.0.1",
"resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-6.0.1.tgz", "integrity": "sha512-
Y38VPSHcqkFrCpFnQ9vuSXmquuv5oXOKpGeT6aGrr3o3Gc9AlVa6JBfUSOCnbxGGZF+/0oolI7KrPuUSztUdU5A==
", "dev": true, "license": "MIT", "dependencies": { "ansi-regex": "^5.0.1" }, "engines": { "node": ">=8" },
"node_modules/strip-ansi": { "version": "7.1.2", "resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-
7.1.2.tgz", "integrity": "sha512-
gmBGslpoQJtgnMAVOVqGZpEz9dyoKTCzy2nfz/n8aIFhN/jCE/rCmcxabB6jOOHV+0WNnylOxaxBQPSvcWklhA==",
"dev": true, "license": "MIT", "dependencies": { "ansi-regex": "^6.0.1" }, "engines": { "node": ">=12" }, "funding": {
"url": "https://github.com/chalk/strip-ansi?sponsor=1" }, "node_modules/strip-ansi-cjs": { "name": "strip-ansi",
"version": "6.0.1", "resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-6.0.1.tgz", "integrity": "sha512-
Y38VPSHcqkFrCpFnQ9vuSXmquuv5oXOKpGeT6aGrr3o3Gc9AlVa6JBfUSOCnbxGGZF+/0oolI7KrPuUSztUdU5A==
", "dev": true, "license": "MIT", "dependencies": { "ansi-regex": "^5.0.1" }, "engines": { "node": ">=8" },
"node_modules/strip-ansi-cjs/node_modules/ansi-regex": { "version": "5.0.1", "resolved":
"https://registry.npmjs.org/ansi-regex/-/ansi-regex-5.0.1.tgz", "integrity": "sha512-
quJQXITSUGL2LH9SUXo8VwsY4soanhgo6LNSm84E1LBcE8s3O0wpdiRzyR9z/ZZJMIMWv37qOOb9pdJIMUEKFQ

```
==", "dev": true, "license": "MIT", "engines": { "node": ">=8" } }, "node_modules/strip-json-comments": { "version":  
"3.1.1", "resolved": "https://registry.npmjs.org/strip-json-comments/-/strip-json-comments-3.1.1.tgz", "integrity":  
"sha512-  
6fPc+R4ihwqP6N/alv2f1gMH8IOVtWQHoqC4yK6oSDVVocumAsfCqjkXnqiYMhmMwS/mEHLp7Vehlt3ql6IEig==",  
"dev": true, "license": "MIT", "engines": { "node": ">=8" }, "funding": { "url":  
"https://github.com/sponsors/sindresorhus" } }, "node_modules/sucrase": { "version": "3.35.0", "resolved":  
"https://registry.npmjs.org/sucrase/-/sucrase-3.35.0.tgz", "integrity": "sha512-  
8EbVDiu9iN/nESwxSxDKe0dunta1GOIHufmSSXxMD2z2/tMZpDMPvXQGSc+ajGo8y2uYUmixaSRUc/QPoQ0GA==  
", "dev": true, "license": "MIT", "dependencies": { "@jridgewell/gen-mapping": "^0.3.2", "commander": "^4.0.0",  
"glob": "^10.3.10", "lines-and-columns": "^1.1.6", "mz": "^2.7.0", "pirates": "^4.0.1", "ts-interface-checker": "^0.1.9" },  
"bin": { "sucrase": "bin/sucrase", "sucrase-node": "bin/sucrase-node" }, "engines": { "node": ">=16 || 14 >=14.17" }  
}, "node_modules/supports-color": { "version": "7.2.0", "resolved": "https://registry.npmjs.org/supports-  
color/-/supports-color-7.2.0.tgz", "integrity": "sha512-  
qpCAvRI9stuOHveKsn7HncJRvv501qlacKzQIO/+Lwxc9+0q2wLyv4Dfvt80/DPn2pqOBsJdDiogXGR9+OvwRw==",  
"dev": true, "license": "MIT", "dependencies": { "has-flag": "^4.0.0" }, "engines": { "node": ">=8" } },  
"node_modules/supports-preserve-symlinks-flag": { "version": "1.0.0", "resolved":  
"https://registry.npmjs.org/supports-preserve-symlinks-flag/-/supports-preserve-symlinks-flag-1.0.0.tgz",  
"integrity": "sha512-  
ot0WnXS9fgdkglcePe6RHNk1WA8+muPa6cSjeR3V8K27q9BB1rTE3R1p7Hv0z1ZyAc8s6Vvv8DlyWf681MA0w==",  
"dev": true, "license": "MIT", "engines": { "node": ">= 0.4" }, "funding": { "url": "https://github.com/sponsors/ljharb"  
} }, "node_modules/tailwind-merge": { "version": "2.6.0", "resolved": "https://registry.npmjs.org/tailwind-  
merge/-/tailwind-merge-2.6.0.tgz", "integrity": "sha512-  
P+Vu1qXfzediirmHOC3xKGAYeZtPcV9g76X+Xg2FD4tYgR71ewMA35Y3sCz3zhiN/dwefRpJX0yBcgwi1fXNQA==",  
"license": "MIT", "funding": { "type": "github", "url": "https://github.com/sponsors/dcastil" } },  
"node_modules/tailwindcss": { "version": "3.4.18", "resolved": "https://registry.npmjs.org/tailwindcss/-/tailwindcss-  
3.4.18.tgz", "integrity": "sha512-  
6A2rnmW5xZMdw11L Yjhcl5846rt9pbL SabY5XPxo+XWdxwZaFEn47Go4NzFiHu9sNNmr/kXivP1vStfvMaK1GQ==",  
"dev": true, "license": "MIT", "dependencies": { "@alloc/quick-lru": "^5.2.0", "arg": "^5.0.2", "chokidar": "^3.6.0",  
"didyoumean": "^1.2.2", "dlv": "^1.1.3", "fast-glob": "^3.3.2", "glob-parent": "^6.0.2", "is-glob": "^4.0.3", "jiti":  
"^1.21.7", "lilconfig": "^3.1.3", "micromatch": "^4.0.8", "normalize-path": "^3.0.0", "object-hash": "^3.0.0",  
"picocolors": "^1.1.1", "postcss": "^8.4.47", "postcss-import": "^15.1.0", "postcss-js": "^4.0.1", "postcss-load-config":  
"^4.0.2 || ^5.0 || ^6.0", "postcss-nested": "^6.2.0", "postcss-selector-parser": "^6.1.2", "resolve": "^1.22.8",  
"sucrase": "^3.35.0" }, "bin": { "tailwind": "lib/cli.js", "tailwindcss": "lib/cli.js" }, "engines": { "node": ">=14.0.0" } },  
"node_modules/tailwindcss-animate": { "version": "1.0.7", "resolved": "https://registry.npmjs.org/tailwindcss-  
animate/-/tailwindcss-animate-1.0.7.tgz", "integrity": "sha512-  
bl6mpH3T713UFxuvDEXLxy/VuFxBk5bbzplh7tXI68mwMokNYd1t9qPBHlNyTwfa4JGC4zP516l1hYYtQ/vspA==",  
"dev": true, "license": "MIT", "peerDependencies": { "tailwindcss": ">=3.0.0 || insiders" } },  
"node_modules/thenify": { "version": "3.3.1", "resolved": "https://registry.npmjs.org/thenify/-/thenify-3.3.1.tgz",  
"integrity": "sha512-  
RVZSIV5IG10Hk3enotrhzv0T9em6cyHBLkH/YAZuKqd8hRkKhSfCGlcp2KUY0EPxndzANBmNllzWPwak+bheSw==",  
"dev": true, "license": "MIT", "dependencies": { "any-promise": "^1.0.0" } }, "node_modules/thenify-all": {  
"version": "1.6.0", "resolved": "https://registry.npmjs.org/thenify-all/-/thenify-all-1.6.0.tgz", "integrity": "sha512-  
RNxQH/qI8/t3thXJDwcstUO4zeqo64+Uy/+sNVRBx4Xn2OX+OZ9oP+iJnNFqplFra2ZUVEKCSa2oVWi3T4uVmA==",  
"dev": true, "license": "MIT", "dependencies": { "thenify": ">= 3.1.0 < 4" }, "engines": { "node": ">=0.8" } },  
"node_modules/tinyglobby": { "version": "0.2.15", "resolved": "https://registry.npmjs.org/tinyglobby/-/tinyglobby-  
0.2.15.tgz", "integrity": "sha512-  
j2Zq4NyQYG5XMST4cbs02Ak8iJUdxRM0XI5QyxXuZOzKOINmWurp3smXu3y5wDcJrptwpSjgXHziQxR0omXljq=  
", "dev": true, "license": "MIT", "dependencies": { "fdir": "^6.5.0", "picomatch": "^4.0.3" }, "engines": { "node":  
">=12.0.0" }, "funding": { "url": "https://github.com/sponsors/SuperchupuDev" } }, "node_modules/to-regexp-
```



```

range": { "version": "5.0.1", "resolved": "https://registry.npmjs.org/to-regexp-range/-/to-regexp-range-5.0.1.tgz",
"integrity": "sha512-
65P7iz6X5yEr1cwcgvQxbblw7Uk3gOy5dldtZ4rDveLqhrdJP+Li/Hx6tyK0NEb+2GCyneCMJiGqrADCSNk8sQ==",
"dev": true, "license": "MIT", "dependencies": { "is-number": "^7.0.0" }, "engines": { "node": ">=8.0" } },
"node_modules/ts-interface-checker": { "version": "0.1.13", "resolved": "https://registry.npmjs.org/ts-interface-
checker/-/ts-interface-checker-0.1.13.tgz", "integrity": "sha512-
Y/arvbn+rrz3JCKI9C4kVNfTfSm2/mEp5FSz5EsZSANGPSIQrpRI5M4PKF+mJnE52jOO90PnPSc3Ur3bTQw0gA==",
"dev": true, "license": "Apache-2.0", "node_modules/tslib": { "version": "2.8.1", "resolved":
"https://registry.npmjs.org/tslib/-/tslib-2.8.1.tgz", "integrity": "sha512-
oJFu94HQb+KVduSUQL7wnpmqnmfLSOA/nAh6b6EH0wCEoK0/mPeXU6c3wKDV83MkOuHPRHtSXKKU99IBazS/
2w==", "license": "0BSD" }, "node_modules/type-check": { "version": "0.4.0", "resolved":
"https://registry.npmjs.org/type-check/-/type-check-0.4.0.tgz", "integrity": "sha512-
XleUoc9uwGXqjWwXaUTZAmzMcFZ5858QA2vux1Ur5xlciXIP+8LnFDgRplU30us6teqdlSkFfu+ae4K79Ooew==",
"dev": true, "license": "MIT", "dependencies": { "prelude-ls": "^1.2.1" }, "engines": { "node": ">= 0.8.0" } },
"node_modules/update-browserslist-db": { "version": "1.1.3", "resolved": "https://registry.npmjs.org/update-
browserslist-db/-/update-browserslist-db-1.1.3.tgz", "integrity": "sha512-
UxhIZQ+QInVdunkDAaiazvvT/+fXL5Osr0JZIjulepYu6Jd7qJtDZjlur0emRIT71EN3ScPoE7gvsulKKNavKw==", "dev":
true, "funding": [ { "type": "opencollective", "url": "https://opencollective.com/browserslist" }, { "type": "tidelift",
"url": "https://tidelift.com/funding/github/npm/browserslist" }, { "type": "github", "url":
"https://github.com/sponsors/ai" } ], "license": "MIT", "dependencies": { "escalade": "^3.2.0", "picocolors": "^1.1.1"
}, "bin": { "update-browserslist-db": "cli.js" }, "peerDependencies": { "browserslist": ">= 4.21.0" } },
"node_modules/uri-js": { "version": "4.4.1", "resolved": "https://registry.npmjs.org/uri-js/-/uri-js-4.4.1.tgz",
"integrity": "sha512-
7rKUyY33Q1yc98pQ1DAmLtwX109F7TfWiW1Ydo8Wl1ii1SeHieeh0HHfPeL2fMXK6z0s8ecKs9frCuLJvndBg==",
"dev": true, "license": "BSD-2-Clause", "dependencies": { "punycode": "^2.1.0" } }, "node_modules/use-callback-
ref": { "version": "1.3.3", "resolved": "https://registry.npmjs.org/use-callback-ref/-/use-callback-ref-1.3.3.tgz",
"integrity": "sha512-
jQL3IRnocaFtu3V00JT0yZ/4QkNWswxijDaCVNZRiRTO3HQDLsdu1ZtmiUvV4yPp+rvWm5j0y0TG/S61cuijTg==",
"license": "MIT", "dependencies": { "tslib": "^2.0.0" }, "engines": { "node": ">=10" }, "peerDependencies": {
"@types/react": " ", "react": "^16.8.0 || ^17.0.0 || ^18.0.0 || ^19.0.0 || ^19.0.0-rc" }, "peerDependenciesMeta": {
"@types/react": { "optional": true } } }, "node_modules/use-sidecar": { "version": "1.1.3", "resolved":
"https://registry.npmjs.org/use-sidecar/-/use-sidecar-1.1.3.tgz", "integrity": "sha512-
Fedw0aZvkhynoPYIA5WXrMCAMm+nSWdZt6lJQ7Ok8S6Q+VsHmHpRWndVRJ8Be0ZbkfPc5LRYH+5XrzXcEeLR
Q==", "license": "MIT", "dependencies": { "detect-node-es": "^1.1.0", "tslib": "^2.0.0" }, "engines": { "node":
">=10" }, "peerDependencies": { "@types/react": "*", "react": "^16.8.0 || ^17.0.0 || ^18.0.0 || ^19.0.0 || ^19.0.0-rc" },
"peerDependenciesMeta": { "@types/react": { "optional": true } } }, "node_modules/util-deprecate": { "version":
"1.0.2", "resolved": "https://registry.npmjs.org/util-deprecate/-/util-deprecate-1.0.2.tgz", "integrity": "sha512-
EPD5q1uXyFxCrLnCc1nHnq3gOa6DZBocAlil2TaSCA7VCJ1UJDMagCzikXNsUYfD1daK//LTEQ8xilbrHtcw==",
"dev": true, "license": "MIT" }, "node_modules/vite": { "version": "7.1.6", "resolved":
"https://registry.npmjs.org/vite/-/vite-7.1.6.tgz", "integrity": "sha512-
SRYIB8t/isTwNn8vMB3MR6E+EQZM/WG1aKmmIUCfDXfvvKfc20ZpamngWHKzAmmu9ppsgxsg4b2I7c90JZudIQ=
=", "dev": true, "license": "MIT", "dependencies": { "esbuild": "^0.25.0", "fdir": "^6.5.0", "picomatch": "^4.0.3",
"postcss": "^8.5.6", "rollup": "^4.43.0", "tinyglobby": "^0.2.15" }, "bin": { "vite": "bin/vite.js" }, "engines": { "node":
"^20.19.0 || >=22.12.0" }, "funding": { "url": "https://github.com/vitejs/vite?sponsor=1" }, "optionalDependencies": {
"fsevents": "~2.3.3" }, "peerDependencies": { "@types/node": "^20.19.0 || >=22.12.0", "jiti": ">=1.21.0", "less":
"^4.0.0", "lightningcss": "^1.21.0", "sass": "^1.70.0", "sass-embedded": "^1.70.0", "stylus": ">=0.54.8", "sugarss":
"^5.0.0", "terser": "^5.16.0", "tsx": "^4.8.1", "yaml": "^2.4.2" }, "peerDependenciesMeta": { "@types/node": {
"optional": true }, "jiti": { "optional": true }, "less": { "optional": true }, "lightningcss": { "optional": true }, "sass": {
"optional": true }, "sass-embedded": { "optional": true }, "stylus": { "optional": true }, "sugarss": { "optional": true

```

```

}, "terser": { "optional": true }, "tsx": { "optional": true }, "yaml": { "optional": true } }, "node_modules/which": {
  "version": "2.0.2", "resolved": "https://registry.npmjs.org/which/-/which-2.0.2.tgz", "integrity": "sha512-
BLI3TI1TW3PvI70I3yq3Y64i+awpwXqsGBYWkkqMtnbXgrMD+yj7rhW0kuEDxzJaYXGjEW5ogapKNMEKNMjibA==",
  "dev": true, "license": "ISC", "dependencies": { "isexe": "^2.0.0" }, "bin": { "node-which": "bin/node-which" },
  "engines": { "node": ">= 8" }, "node_modules/word-wrap": { "version": "1.2.5", "resolved":
  "https://registry.npmjs.org/word-wrap/-/word-wrap-1.2.5.tgz", "integrity": "sha512-
BN22B5eaMMI9UMtjrGd5g5eCYPpCPDUy0FJXbYsaT5zYxjFOckS53SQDE3pWkVoWpHXVb3BrYcEN4Twa55B5cA
==", "dev": true, "license": "MIT", "engines": { "node": ">=0.10.0" }, "node_modules/wrap-ansi": { "version":
  "8.1.0", "resolved": "https://registry.npmjs.org/wrap-ansi/-/wrap-ansi-8.1.0.tgz", "integrity": "sha512-
si7QWI6zUMq56bESFvagtzmzMdGOtoxfR+Sez11Mobfc7tm+VkUckk9bW2UeffTGVUbOksxmSw0AA2gs8g71NCQ=
=", "dev": true, "license": "MIT", "dependencies": { "ansi-styles": "^6.1.0", "string-width": "^5.0.1", "strip-ansi":
  "^7.0.1" }, "engines": { "node": ">=12" }, "funding": { "url": "https://github.com/chalk/wrap-ansi?sponsor=1" } },
  "node_modules/wrap-ansi-cjs": { "name": "wrap-ansi", "version": "7.0.0", "resolved":
  "https://registry.npmjs.org/wrap-ansi/-/wrap-ansi-7.0.0.tgz", "integrity": "sha512-
YVGJl2kamLSTxw6NsZjoBxfSwsn0ycdesmc4p+Q21c5zPuZ1pl+NfxVdxPtdHvmNVOQ6XSYG4AUtyt/Fi7D16Q==",
  "dev": true, "license": "MIT", "dependencies": { "ansi-styles": "^4.0.0", "string-width": "^4.1.0", "strip-ansi":
  "^6.0.0" }, "engines": { "node": ">=10" }, "funding": { "url": "https://github.com/chalk/wrap-ansi?sponsor=1" } },
  "node_modules/wrap-ansi-cjs/node_modules/ansi-regex": { "version": "5.0.1", "resolved":
  "https://registry.npmjs.org/ansi-regex/-/ansi-regex-5.0.1.tgz", "integrity": "sha512-
quJQXITSUGL2LH9SUXo8VwsY4soanhgo6LNSm84E1LBcE8s3O0wpdiRzyR9z/ZZJMIMWv37qOOB9pdJIMUEKFK
==", "dev": true, "license": "MIT", "engines": { "node": ">=8" }, "node_modules/wrap-ansi-
cjs/node_modules/emoji-regex": { "version": "8.0.0", "resolved": "https://registry.npmjs.org/emoji-regex/-/emoji-
regex-8.0.0.tgz", "integrity": "sha512-
MSjYzcWNOA0ewAHpz0MxpYFvwg6yjj1NG3xteoqz644VCo/RPgnr1/GGt+ic3iJTzQ8Eu3TdM14SawnVUmGE6A==",
  "dev": true, "license": "MIT" }, "node_modules/wrap-ansi-cjs/node_modules/string-width": { "version": "4.2.3",
  "resolved": "https://registry.npmjs.org/string-width/-/string-width-4.2.3.tgz", "integrity": "sha512-
wKyQRQpjJ0sIp62ErSZdGsjMJWsap5oRNihHhu6G7JVO/9jIB6UyevL+tXuOqrng8j/cxKTWYwUwvSTriiZz/g==",
  "dev": true, "license": "MIT", "dependencies": { "emoji-regex": "^8.0.0", "is-fullwidth-code-point": "^3.0.0",
  "strip-ansi": "^6.0.1" }, "engines": { "node": ">=8" }, "node_modules/wrap-ansi-cjs/node_modules/strip-ansi": {
  "version": "6.0.1", "resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-6.0.1.tgz", "integrity": "sha512-
Y38VPSHcqKfRcPfnQ9vuSXMquuv5oXOKpGeT6aGrr3o3Gc9AlVa6JBfUSOCnbxGGZF+/0oolI7KrPuUSztUdU5A==",
  "dev": true, "license": "MIT", "dependencies": { "ansi-regex": "^5.0.1" }, "engines": { "node": ">=8" } },
  "node_modules/wrap-ansi/node_modules/ansi-styles": { "version": "6.2.3", "resolved":
  "https://registry.npmjs.org/ansi-styles/-/ansi-styles-6.2.3.tgz", "integrity": "sha512-
4Dj6M28JB+oAH8kFkTLUo+a2jwOFkuqb3yucU0CANcRRUbxS0cP0nZYCGjcc3BNXwRIsUVmDGgzawme7zvJHvg
==", "dev": true, "license": "MIT", "engines": { "node": ">=12" }, "funding": { "url": "https://github.com/chalk/ansi-
styles?sponsor=1" } }, "node_modules/yallist": { "version": "3.1.1", "resolved":
  "https://registry.npmjs.org/yallist/-/yallist-3.1.1.tgz", "integrity": "sha512-
a4UGQaWPH59mOXUYnAG2ewncQS4i4F43Tv3JoAM+s2VDAmS9NsK8GpDMLrCHPkFT7h3K6TOoUNn2pb7Ro
Xx4g==", "dev": true, "license": "ISC" }, "node_modules/yocto-queue": { "version": "0.1.0", "resolved":
  "https://registry.npmjs.org/yocto-queue/-/yocto-queue-0.1.0.tgz", "integrity": "sha512-
rVksvsnnNCdJ/ohGc6xgPwyN8eheCxsilM8mxuE/t/mOVqJewPuO1miLpTHQiRgTKCLexL4MeAFVagts7HmNZ2Q==",
  "dev": true, "license": "MIT", "engines": { "node": ">=10" }, "funding": { "url":
  "https://github.com/sponsors/sindresorhus" } } } }

===== FILE: frontend/postcss.config.js ===== export default { plugins: { tailwindcss: {}, autoprefixer: {}, }, }

```

===== FILE: frontend/public/vite.svg =====



```
===== FILE: frontend/src/App.jsx ===== import { useState } from 'react' import { Routes, Route, Navigate, Link,
useNavigate } from 'react-router-dom' import { useAuth } from './AuthContext.jsx' import AccountHome from
'./features/groups/AccountHome.jsx' import GroupDetails from './features/groups/GroupDetails.jsx' import { Button
} from './components/ui/button.jsx' import { Card, CardHeader, CardTitle, CardContent, CardFooter } from
'./components/ui/card.jsx' import { Input } from './components/ui/input.jsx' import { Label } from
'./components/ui/label.jsx'

function ProtectedRoute({ children }) { const { isAuthenticated } = useAuth() if (!isAuthenticated) return return
children }

function Navbar() { const { isAuthenticated, logout, name } = useAuth() const navigate = useNavigate()

function handleLogout() { logout() navigate('/') }

return (

MyCount
{isAuthenticated ? (
  Dashboard {name} Logout
) : (
  Login Sign up
)}
)}

function Landing() { return (
```

Split expenses the simple way

Track, share, and settle group expenses with ease using a streamlined, modern interface powered by shadcn/ui.

```
Get started Log in
)}

function Login() { const { login, isLoading } = useAuth() const navigate = useNavigate() const [error, setError] =
useState(null)

async function onSubmit(e) { e.preventDefault() const form = new FormData(e.currentTarget) const email =
form.get('email') const pw = form.get('pw') try { setError(null) await login({ email, pw }) navigate('/account') }
catch (err) { setError(err.message || 'Login failed') } }

return (

Log in
Email name@example.com
Password .....
{error ?

{error}

: null} {isLoading ? 'Loading...' : 'Log in'}

No account?{' '} Sign up

)}
```

```
function Signup() { const { signup, isLoading } = useAuth() const navigate = useNavigate() const [error, setError] =
useState(null)
```

```
async function onSubmit(e) { e.preventDefault() const form = new FormData(e.currentTarget) const name =
form.get('name') const email = form.get('email') const pw = form.get('pw') try { setError(null) await signup({ name,
email, pw }) navigate('/account') } catch (err) { setError(err.message || 'Signup failed') } }
```

```
return (
```

```
Create your account
```

```
Name 
```

```
Email 
```

```
Password 
```

```
{error ?
```

```
{error}
```

```
: null} 
```

```
Already have an account?{' '} Log in
```

```
) }
```

```
export default function App() { return (
```

```
<Route path="/" element={} /> <Route path="/login" element={} /> <Route path="/signup" element={} /> <Route
path="/account" element={} /> <Route path="/account/:groupName" element={} /> <Route path="*" element={} />
) }
```

```
===== FILE: frontend/src/AuthContext.jsx ===== import { createContext, useContext, useEffect, useMemo,
useState } from 'react' import { loginUser, signupUser, logout as apiLogout } from './services/api'
```

```
const AuthContext = createContext(null)
```

```
export function AuthProvider({ children }) { const [token, setToken] = useState(() => localStorage.getItem('token'))
const [userId, setUserId] = useState(() => localStorage.getItem('userId')) const [name, setUserName] =
useState(() => localStorage.getItem('name')) const [isLoading, setIsLoading] = useState(false)
```

```
useEffect(() => { if (token) localStorage.setItem('token', token) else localStorage.removeItem('token') }, [token])
```

```
useEffect(() => { if (userId) localStorage.setItem('userId', userId) else localStorage.removeItem('userId') },
[userId])
```

```
async function login(credentials) { setIsLoading(true) try { const data = await loginUser(credentials) if
(data?.access_token) setToken(data.access_token) if (data?.user?.id) setUserId(String(data.user.id)) if
(data?.user?.name) setUserName(String(data.user.name)) return data } finally { setIsLoading(false) } }
```

```
async function signup(payload) { setIsLoading(true) try { const data = await signupUser(payload) if
(data?.access_token) setToken(data.access_token) if (data?.user?.id) setUserId(String(data.user.id)) if
(data?.user?.name) setUserName(String(data.user.name)) return data } finally { setIsLoading(false) } }
```

```
function logout() { apiLogout() setToken(null) setUserId(null) }
```

```
const value = useMemo( () => ({ token, userId, name, isAuthenticated: Boolean(token), isLoading, login, signup,
logout }), [token, userId, name, isLoading] )
```

```
return ( <AuthContext.Provider value={value}> {children} </AuthContext.Provider> ) }
```

```
// eslint-disable-next-line react-refresh/only-export-components export function useAuth() { const ctx =
useContext(AuthContext) if (!ctx) throw new Error('useAuth must be used within AuthProvider') return ctx }
```

```
===== FILE: frontend/src/assets/react.svg =====
```



```
===== FILE: frontend/src/components/ui/badge.jsx ===== import { cva } from "class-variance-authority" import {
cn } from "../../lib/utils"
```

```
const badgeVariants = cva( "inline-flex items-center rounded-full border px-2.5 py-0.5 text-xs font-semibold
transition-colors focus:outline-none focus:ring-2 focus:ring-ring focus:ring-offset-2", { variants: { variant: {
default: "border-transparent bg-primary text-primary-foreground hover:bg-primary/80", secondary: "border-
transparent bg-secondary text-secondary-foreground hover:bg-secondary/80", destructive: "border-transparent
bg-destructive text-destructive-foreground hover:bg-destructive/80", outline: "text-foreground" } },
defaultVariants: { variant: "default" } } )
```

```
function Badge({ className, variant, ...props }) { return <div className={cn(badgeVariants({ variant }),
className)} {...props} /> }
```

```
// eslint-disable-next-line react-refresh/only-export-components export { Badge, badgeVariants }
```

```
===== FILE: frontend/src/components/ui/button.jsx ===== import * as React from "react" import { Slot } from
"@radix-ui/react-slot" import { cva } from "class-variance-authority"
```

```
import { cn } from "../../lib/utils"
```

```
const buttonVariants = cva( "inline-flex items-center justify-center whitespace-nowrap rounded-md text-sm font-
medium transition-colors focus-visible:outline-none focus-visible:ring-2 focus-visible:ring-ring focus-visible:ring-
offset-2 disabled:pointer-events-none disabled:opacity-50 ring-offset-background", { variants: { variant: { default:
"bg-primary text-primary-foreground hover:bg-primary/90", secondary: "bg-secondary text-secondary-
foreground hover:bg-secondary/80", outline: "border border-input bg-transparent hover:bg-accent hover:text-
accent-foreground", ghost: "hover:bg-accent hover:text-accent-foreground", destructive: "bg-destructive text-
destructive-foreground hover:bg-destructive/90", link: "underline-offset-4 hover:underline text-primary" } }, size: {
default: "h-10 px-4 py-2", sm: "h-9 rounded-md px-3", lg: "h-11 rounded-md px-8", icon: "h-10 w-10" } },
defaultVariants: { variant: "default", size: "default" } } )
```

```
const Button = React.forwardRef(({ className, variant, size, asChild = false, ...props }, ref) => { const Comp =
asChild ? Slot : "button" return ( <Comp ref={ref} className={cn(buttonVariants({ variant, size, className })} )
{...props} /> ) } ) Button.displayName = "Button"
```

```
// eslint-disable-next-line react-refresh/only-export-components export { Button, buttonVariants }
```

```
===== FILE: frontend/src/components/ui/card.jsx ===== import * as React from "react"
```

```
import { cn } from "../../lib/utils"
```

```
const Card = React.forwardRef(({ className, ...props }, ref) => (
```

```
)) Card.displayName = "Card"
```

```
const CardHeader = React.forwardRef(({ className, ...props }, ref) => (
```

```
)) CardHeader.displayName = "CardHeader"
```

```

const CardTitle = React.forwardRef(({ className, ...props }, ref) => (

)) CardTitle.displayName = "CardTitle"

const CardDescription = React.forwardRef(({ className, ...props }, ref) => (

)) CardDescription.displayName = "CardDescription"

const CardContent = React.forwardRef(({ className, ...props }, ref) => (

)) CardContent.displayName = "CardContent"

const CardFooter = React.forwardRef(({ className, ...props }, ref) => (

)) CardFooter.displayName = "CardFooter"

export { Card, CardHeader, CardFooter, CardTitle, CardDescription, CardContent }

===== FILE: frontend/src/components/ui/checkbox.jsx ===== import * as React from "react" import * as
CheckboxPrimitive from "@radix-ui/react-checkbox" import { Check } from "lucide-react"

import { cn } from "../../lib/utils"

const Checkbox = React.forwardRef(({ className, ...props }, ref) => ( <CheckboxPrimitive.Root ref={ref}
className={cn( "peer h-4 w-4 shrink-0 rounded-sm border border-primary/50 focus-visible:outline-none focus-
visible:ring-2 focus-visible:ring-ring focus-visible:ring-offset-2 disabled:cursor-not-allowed disabled:opacity-50
data-[state=checked]:bg-primary data-[state=checked]:text-primary-foreground", className )} {...props}

  <CheckboxPrimitive.Indicator className="flex items-center justify-center text-
current">
    <Check className="h-3 w-3" />
  </CheckboxPrimitive.Indicator>

</CheckboxPrimitive.Root> )) Checkbox.displayName = CheckboxPrimitive.Root.displayName

export { Checkbox }

===== FILE: frontend/src/components/ui/dialog.jsx ===== import * as DialogPrimitive from "@radix-ui/react-
dialog" import { X } from "lucide-react" import { forwardRef } from "react"

import { cn } from "../../lib/utils" import { buttonVariants } from "../button"

const Dialog = DialogPrimitive.Root

const DialogTrigger = DialogPrimitive.Trigger

const DialogClose = DialogPrimitive.Close

const DialogPortal = DialogPrimitive.Portal

const DialogOverlay = forwardRef(({ className, ...props }, ref) => ( <DialogPrimitive.Overlay ref={ref} className=
{cn( "fixed inset-0 z-50 bg-black/70 backdrop-blur-sm data-[state=open]:animate-in data-
[state=closed]:animate-out data-[state=closed]:fade-out data-[state=open]:fade-in", className )} {...props} /> ))
DialogOverlay.displayName = DialogPrimitive.Overlay.displayName

```

```

const DialogContent = forwardRef(({ className, children, ...props }, ref) => ( <DialogPrimitive.Content ref={ref}
className={cn( "fixed left-[50%] top-[50%] z-50 grid w-full max-w-xl translate-x-[-50%] translate-y-[-50%]
gap-4 border bg-popover p-6 shadow-lg duration-200 data-[state=open]:animate-in data-
[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0 data-
[state=closed]:zoom-out-95 data-[state=open]:zoom-in-95 data-[state=closed]:slide-out-to-left-1/2 data-
[state=closed]:slide-out-to-top-[48%] data-[state=open]:slide-in-from-left-1/2 data-[state=open]:slide-in-from-
top-[48%]", className )} {...props} > {children} <DialogPrimitive.Close className={cn( buttonVariants({ variant:
"ghost", size: "icon" }), "absolute right-4 top-4 h-8 w-8 p-0 text-muted-foreground hover:text-foreground" )}
aria-label="Close dialog" > </DialogPrimitive.Close> </DialogPrimitive.Content> )) DialogContent.displayName =
DialogPrimitive.Content.displayName

const DialogHeader = ({ className, ...props }) => (

) DialogHeader.displayName = "DialogHeader"

const DialogFooter = ({ className, ...props }) => (

) DialogFooter.displayName = "DialogFooter"

const DialogTitle = forwardRef(({ className, ...props }, ref) => ( <DialogPrimitive.Title ref={ref} className=
{cn("text-lg font-semibold leading-none tracking-tight", className)} {...props} /> )) DialogTitle.displayName =
DialogPrimitive.Title.displayName

const DialogDescription = forwardRef(({ className, ...props }, ref) => ( <DialogPrimitive.Description ref={ref}
className={cn("text-sm text-muted-foreground", className)} {...props} /> )) DialogDescription.displayName =
DialogPrimitive.Description.displayName

export { Dialog, DialogPortal, DialogOverlay, DialogTrigger, DialogContent, DialogHeader, DialogFooter, DialogTitle,
DialogDescription, DialogClose, }

===== FILE: frontend/src/components/ui/input.jsx ===== import * as React from "react"

import { cn } from "../../lib/utils"

const Input = React.forwardRef(({ className, type = "text", ...props }, ref) => { return ( <input type={type}
className={cn( "flex h-10 w-full rounded-md border border-input bg-transparent px-3 py-2 text-sm
placeholder:text-muted-foreground shadow-sm transition-colors focus-visible:outline-none focus-visible:ring-2
focus-visible:ring-ring focus-visible:ring-offset-2 disabled:cursor-not-allowed disabled:opacity-50", className )}
ref={ref} {...props} /> ) }) Input.displayName = "Input"

export { Input }

===== FILE: frontend/src/components/ui/label.jsx ===== import * as React from "react" import * as LabelPrimitive
from "@radix-ui/react-label" import { cva } from "class-variance-authority"

import { cn } from "../../lib/utils"

const labelVariants = cva( "text-sm font-medium leading-none peer-disabled:cursor-not-allowed peer-
disabled:opacity-70" )

const Label = React.forwardRef(({ className, ...props }, ref) => ( <LabelPrimitive.Root ref={ref} className=
{cn(labelVariants(), className)} {...props} /> )) Label.displayName = LabelPrimitive.Root.displayName

export { Label }

```

```
===== FILE: frontend/src/components/ui/select.jsx ===== import * as React from "react" import * as
SelectPrimitive from "@radix-ui/react-select" import { Check, ChevronDown, ChevronUp } from "lucide-react"

import { cn } from "../../lib/utils"
```

```
const Select = SelectPrimitive.Root
```

```
const SelectGroup = SelectPrimitive.Group
```

```
const SelectValue = SelectPrimitive.Value
```

```
const SelectTrigger = React.forwardRef(({ className, children, ...props }, ref) => ( <SelectPrimitive.Trigger ref=
{ref} className={cn( "flex h-10 w-full items-center justify-between rounded-md border border-input bg-
transparent px-3 py-2 text-sm shadow-sm ring-offset-background placeholder:text-muted-foreground
focus:outline-none focus:ring-2 focus:ring-ring focus:ring-offset-2 disabled:cursor-not-allowed disabled:opacity-
50", className )} {...props}
```

```
  {children}
  <SelectPrimitive.Icon asChild>
    <ChevronDown className="h-4 w-4 opacity-50" />
  </SelectPrimitive.Icon>
```

```
</SelectPrimitive.Trigger> )) SelectTrigger.displayName = SelectPrimitive.Trigger.displayName
```

```
const SelectContent = React.forwardRef(({ className, children, position = "popper", ...props }, ref) => (
<SelectPrimitive.Portal> <SelectPrimitive.Content ref={ref} className={cn( "relative z-50 min-w-[8rem]
overflow-hidden rounded-md border bg-popover text-popover-foreground shadow-md data-
[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-
[state=open]:fade-in-0 data-[state=closed]:zoom-out-95 data-[state=open]:zoom-in-95 data-
[side=bottom]:slide-in-from-top-2 data-[side=left]:slide-in-from-right-2 data-[side=right]:slide-in-from-left-2
data-[side=top]:slide-in-from-bottom-2", position === "popper" && "data-[side=bottom]:translate-y-1 data-
[side=top]:-translate-y-1", className )} position={position} {...props} > <SelectPrimitive.ScrollUpButton
className="flex cursor-default items-center justify-center py-1"> </SelectPrimitive.ScrollUpButton>
<SelectPrimitive.Viewport className="p-1"> {children} </SelectPrimitive.Viewport>
<SelectPrimitive.ScrollDownButton className="flex cursor-default items-center justify-center py-1">
</SelectPrimitive.ScrollDownButton> </SelectPrimitive.Content> </SelectPrimitive.Portal> ))
```

```
SelectContent.displayName = SelectPrimitive.Content.displayName
```

```
const SelectItem = React.forwardRef(({ className, children, ...props }, ref) => ( <SelectPrimitive.Item ref={ref}
className={cn( "relative flex w-full cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm
outline-none focus:bg-accent focus:text-accent-foreground data-[disabled]:pointer-events-none data-
[disabled]:opacity-50", className )} {...props}
```

```
  <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
    <SelectPrimitive.ItemIndicator>
      <Check className="h-4 w-4" />
    </SelectPrimitive.ItemIndicator>
  </span>

  <SelectPrimitive.ItemText>{children}</SelectPrimitive.ItemText>
```

```

</SelectPrimitive.Item> )) SelectItem.displayName = SelectPrimitive.Item.displayName

const SelectSeparator = React.forwardRef(({ className, ...props }, ref) => ( <SelectPrimitive.Separator ref={ref}
className={cn("mx-1 my-1 h-px bg-muted", className)} {...props} /> )) SelectSeparator.displayName =
SelectPrimitive.Separator.displayName

export { Select, SelectGroup, SelectValue, SelectTrigger, SelectContent, SelectItem, SelectSeparator, }

===== FILE: frontend/src/components/ui/tabs.jsx ===== import * as TabsPrimitive from "@radix-ui/react-tabs"
import { forwardRef } from "react"

import { cn } from "../../lib/utils"

const Tabs = TabsPrimitive.Root

const TabsList = forwardRef(({ className, ...props }, ref) => ( <TabsPrimitive.List ref={ref} className={cn(
"inline-flex h-10 items-center justify-center rounded-md bg-muted p-1 text-muted-foreground", className )}
...props /> )) TabsList.displayName = TabsPrimitive.List.displayName

const TabsTrigger = forwardRef(({ className, ...props }, ref) => ( <TabsPrimitive.Trigger ref={ref} className={cn(
"inline-flex items-center justify-center whitespace-nowrap rounded-sm px-3 py-1.5 text-sm font-medium
transition-all focus-visible:outline-none focus-visible:ring-2 focus-visible:ring-ring focus-visible:ring-offset-2
disabled:pointer-events-none disabled:opacity-50 data-[state=active]:bg-background data-[state=active]:text-
foreground", className )} {...props} /> )) TabsTrigger.displayName = TabsPrimitive.Trigger.displayName

const TabsContent = forwardRef(({ className, ...props }, ref) => ( <TabsPrimitive.Content ref={ref} className=
{cn( "mt-4 focus-visible:outline-none focus-visible:ring-2 focus-visible:ring-ring focus-visible:ring-offset-2",
className )} {...props} /> )) TabsContent.displayName = TabsPrimitive.Content.displayName

export { Tabs, TabsList, TabsTrigger, TabsContent }

===== FILE: frontend/src/features/groups/AccountHome.jsx ===== import { useEffect, useState } from 'react'
import { useNavigate } from 'react-router-dom' import { useAuth } from '../../AuthContext.jsx' import { createGroup,
joinGroup, getUserGroups, getGroupDetails } from '../../services/api.js' import { saveGroupDetails,
rememberGroupId, syncGroupSummaries } from './groupStorage.js' import { Button } from
'../../components/ui/button.jsx' import { Dialog, DialogContent, DialogHeader, DialogTitle, DialogDescription,
DialogFooter, DialogClose, } from '../../components/ui/dialog.jsx' import { Input } from '../../components/ui/input.jsx'
import { Label } from '../../components/ui/label.jsx' import { Card } from '../../components/ui/card.jsx' import { cn }
from '../../lib/utils.js'

function EmojiPicker({ selectedEmoji, onEmojiSelect }) { const emojis = [ { name: 'Party', code: '🎉' }, { name:
'Money', code: '💰' }, { name: 'Food', code: '🍕' }, { name: 'Travel', code: '✈️' }, { name: 'Home', code: '🏠' }, {
name: 'Car', code: '🚗' }, { name: 'Gift', code: '🎁' }, { name: 'Heart', code: '❤️' }, { name: 'Star', code: '⭐' }, {
name: 'Fire', code: '🔥' }, { name: 'Lightning', code: '⚡' }, { name: 'Rocket', code: '🚀' } ]

return (

{emojis.map((emoji) => ( <Button key={emoji.code} type="button" variant={selectedEmoji === emoji.code ?
'default' : 'outline'} size="sm" className={cn( 'h-10 w-full px-0 text-xl', selectedEmoji === emoji.code &&
'shadow-lg' )} onClick={() => onEmojiSelect(emoji.code)} title={emoji.name} > {emoji.code} )} )}

) }

```

```
function CreateGroupDialog({ open, onOpenChange, onSubmit }) { const [name, setName] = useState("") const
[groupPw, setGroupPw] = useState("") const [emoji, setEmoji] = useState("") const [isLoading, setIsLoading] =
useState(false) const [errorMessage, setErrorMessage] = useState(null)

useEffect(() => { if (!open) { setErrorMessage(null) setIsLoading(false) } }, [open])

async function handleSubmit(e) { e.preventDefault() setIsLoading(true) try { setErrorMessage(null) await
onSubmit({ name, group_pw: groupPw, emoji: emoji || null }) setName("") setGroupPw("") setEmoji("") } catch (err) {
setErrorMessage(err.message || 'Failed to create group') } finally { setIsLoading(false) } }

return (
  Group name <Input id="group-name" type="text" value={name} onChange={(e) => setName(e.target.value)}
placeholder="Weed"
  Group password <Input id="group-pw" type="password" value={groupPw} onChange={(e) =>
setGroupPw(e.target.value)} placeholder="Secure password" required />
  Choose an emoji (optional)
  {errorMessage ?
    {errorMessage}
    : null}
  <button type="button" value="Cancel" />
  <button type="button" value={isLoading ? 'Creating...' : 'Create group'} />
)
```

```
function JoinGroupDialog({ open, onOpenChange, onSubmit }) { const [inviteLink, setInviteLink] = useState("")
const [groupName, setGroupName] = useState("") const [groupPw, setGroupPw] = useState("") const [isLoading,
setIsLoading] = useState(false) const [errorMessage, setErrorMessage] = useState(null)

useEffect(() => { if (!open) { setErrorMessage(null) setIsLoading(false) } }, [open])

async function handleSubmit(e) { e.preventDefault() setIsLoading(true) try { const trimmedLink = inviteLink.trim()
```

```
    if (trimmedLink) {
      setErrorMessage(null)
      await onSubmit({
        link_auth: trimmedLink,
      })
      setInviteLink('')
      setGroupName('')
      setGroupPw('')
      return
    }

    if (groupName === '' || groupPw === '') {
      setErrorMessage('Provide either an invite link or a group name and
password')
      setIsLoading(false)
      return
    }

    setErrorMessage(null)
    await onSubmit({
      pw_auth: {
```

```

        group_name: groupName,
        group_pw: groupPw,
      },
    })
    setInviteLink('')
    setGroupName('')
    setGroupPw('')
  } catch (err) {
    setErrorMessage(err.message || 'Failed to join group')
  } finally {
    setIsLoading(false)
  }
}

```

```

}

```

```

return (

```

Group invite link <Input Join existing group Enter the name and password shared with you.

```

setInviteLink(e.target.

```

or join with group name & password

Group name <Input id="join-name" type="text" value={groupName} onChange={(e) =>

```

setGroupName(e.target.value)} placeholder="Weekend trip" required={!inviteLink.trim()} />

```

Group password <Input id="join-pw" type="password" value={groupPw} onChange={(e) =>

```

setGroupPw(e.target.value)} placeholder="Password" required={!inviteLink.trim()} />

```

```

{errorMessage ?

```

```

{errorMessage}

```

```

: null} <Cancel {isLoading ? 'Joining...' : 'Join group'}

```

```

) }

```

```

export default function AccountHome() { const { name } = useAuth() const navigate = useNavigate() const
[groups, setGroups] = useState([]) const [isLoading, setIsLoading] = useState(true) const [error, setError] =
useState(null) const [isCreateOpen, setIsCreateOpen] = useState(false) const [isJoinOpen, setIsJoinOpen] =
useState(false)

```

```

useEffect(() => { loadGroups() }, [])

```

```

async function loadGroups() { setIsLoading(true) setError(null) try { const result = await getUserGroups() const list
= Array.isArray(result) ? result : result ? [result] : [] setGroups(list) syncGroupSummaries(list) } catch (err) {
setError(err.message || 'Failed to load groups') } finally { setIsLoading(false) } }

```

```

async function handleNavigateToGroup(group) { if (!group?.id || !group?.name) return

```

```

rememberGroupId(group.name, group.id) try { const fullGroup = await getGroupDetails(group.id) if (fullGroup) {
saveGroupDetails(fullGroup) navigate( /account/${encodeURIComponent( fullGroup.name ) }, { state: {
groupId: fullGroup.id, group: fullGroup, } }) return } } catch (err) { console.error('Failed to load full group details',
err) }

```

```

navigate(`/account/${encodeURIComponent(group.name)}`, {
  state: {
    groupId: group.id,

```

```
}  
})
```

```
}  
  
async function handleCreateGroup(payload) { const group = await createGroup(payload) if (group) {  
  saveGroupDetails(group) handleNavigateToGroup(group) setIsCreateOpen(false) await loadGroups() }}  
  
async function handleJoinGroup(payload) { const group = await joinGroup(payload) if (group) {  
  saveGroupDetails(group) handleNavigateToGroup(group) setIsJoinOpen(false) await loadGroups() }}  
  
return (
```

Your groups

Logged in as {name}

```
<div className="flex flex-wrap gap-3">  
  <Button onClick={() => setIsCreateOpen(true)}>Create new group</Button>  
  <Button variant="outline" onClick={() => setIsJoinOpen(true)}>Join existing  
group</Button>  
</div>  
  
{error && (  
  <div className="rounded-lg border border-destructive/40 bg-destructive/10  
p-4 text-sm text-destructive">  
    {error}  
  </div>  
)}  
  
{!error && isLoading && (  
  <div className="rounded-lg border border-dashed border-muted/50 bg-muted/20  
p-8 text-center text-sm text-muted-foreground">  
    Loading groups...  
  </div>  
)}  
  
{!error && !isLoading && groups.length === 0 && (  
  <div className="rounded-lg border border-dashed border-muted/50 bg-muted/20  
p-8 text-center text-sm text-muted-foreground">  
    No groups yet.  
  </div>  
)}  
  
{!error && !isLoading && groups.length > 0 && (  
  <div className="grid gap-4 sm:grid-cols-2 lg:grid-cols-3">  
    {groups.map((group) => (  
      <Card key={group.id} className="transition hover:border-primary/70  
hover:shadow-lg">  
        <button  
          type="button"  
          onClick={() => handleNavigateToGroup(group)}  
        >
```

```

        className="flex w-full items-center gap-4 rounded-xl px-4 py-5
text-left"
      >
        <span className="text-3xl">{group.emoji || '👥'}</span>
        <span className="text-lg font-medium">{group.name}</span>
      </button>
    </Card>
  )}
</div>
)}

<CreateGroupDialog
  open={isCreateOpen}
  onOpenChange={setIsCreateOpen}
  onSubmit={handleCreateGroup}
/>

<JoinGroupDialog
  open={isJoinOpen}
  onOpenChange={setIsJoinOpen}
  onSubmit={handleJoinGroup}
/>
</section>

```

```

)}

```

```

===== FILE: frontend/src/features/groups/AddExpenseModal.jsx =====
import { useEffect, useMemo, useState }
from 'react' import { createGroupExpense, updateGroupExpense } from '../services/api.js' import { Dialog,
DialogContent, DialogHeader, DialogTitle, DialogDescription, DialogFooter, DialogClose, } from
'../components/ui/dialog.jsx' import { Button } from '../components/ui/button.jsx' import { Input } from
'../components/ui/input.jsx' import { Label } from '../components/ui/label.jsx' import { Select, SelectContent,
SelectItem, SelectTrigger, SelectValue, } from '../components/ui/select.jsx' import { Checkbox } from
'../components/ui/checkbox.jsx' import { cn } from '../lib/utlis.js'

```

```

const roundTwo = (value) => Math.round((value + Number.EPSILON) * 100) / 100

```

```

function buildInitialSplits(members) { return members.map((member) => ({ id: member.id, name: member.name,
isSelected: true, isManual: false, amount: 0, })) }

```

```

function buildSplitsFromExpense(members, expense) { const splitMap = new Map() if (expense?.splits) { for
(const split of expense.splits) { if (split?.user?.id != null) { splitMap.set(split.user.id, roundTwo(split.amount ?? 0)) }
} }

```

```

const participatingIds = Array.from(splitMap.keys()) const hasParticipants = participatingIds.length > 0 const total
= hasParticipants ? roundTwo(participatingIds.reduce((sum, id) => sum + (splitMap.get(id) ?? 0), 0)) : 0 const
evenShare = hasParticipants ? roundTwo(total / participatingIds.length) : 0 const isEvenSplit = hasParticipants &&
participatingIds.every((id) => Math.abs((splitMap.get(id) ?? 0) - evenShare) < 0.02)

```

```

return members.map((member) => { if (!splitMap.has(member.id)) { return { id: member.id, name: member.name,
isSelected: false, isManual: false, amount: 0, } } const amount = splitMap.get(member.id) ?? 0 return { id:
member.id, name: member.name, isSelected: true, isManual: !isEvenSplit, amount, } })

```

```
export default function AddExpenseModal({ groupId, members, currentUserId, onClose, onSuccess, mode =
'create', expense = null, }) { const [description, setDescription] = useState("") const [amountInput,
setAmountInput] = useState("") const [paidById, setPaidById] = useState(null) const [splits, setSplits] =
useState(() => buildInitialSplits(members)) const [error, setError] = useState(null) const [isSubmitting,
setIsSubmitting] = useState(false)
```

```
const amountValue = useMemo(() => { const parsed = parseFloat(amountInput) return Number.isFinite(parsed) ?
parsed : NaN }, [amountInput])
```

```
useEffect(() => { const defaultPayer = () => { if (currentUserId && members.some((m) => m.id ===
currentUserId)) return currentUserId return members[0]?.id ?? null }
```

```
if (mode === 'edit' && expense) {
  const initialSplits = buildSplitsFromExpense(members, expense)
  setDescription(expense.description ?? '')
  setAmountInput(expense.amount !== null ? String(expense.amount) : '')
  const payerId = expense.paid_by?.id ?? defaultPayer()
  setPaidById(payerId)
  setSplits(initialSplits)
} else {
  const initialSplits = buildInitialSplits(members)
  setDescription('')
  setAmountInput('')
  setPaidById(defaultPayer())
  setSplits(initialSplits)
}
```

```
}, [mode, expense, members, currentUserId])
```

```
useEffect(() => { setSplits((prev) => redistributeSplits(prev, amountValue)) }, [amountValue])
```

```
useEffect(() => { if (!Number.isFinite(amountValue) || amountValue <= 0) { setError(null) return }
```

```
const selectedSplits = splits.filter((split) => split.isSelected)
const totalAllocated = roundTwo(selectedSplits.reduce((acc, split) => acc +
split.amount, 0))
const manualTotal = selectedSplits
  .filter((split) => split.isManual)
  .reduce((acc, split) => acc + split.amount, 0)

if (manualTotal - amountValue > 0.009) {
  setError('Manual splits exceed the total amount.')
  return
}

if (selectedSplits.length === 0) {
  setError('Select at least one participant.')
  return
}

if (Math.abs(totalAllocated - amountValue) > 0.01) {
```

```

    setError('Splits must add up to the total amount.')
    return
  }

  setError(null)

```

```

}, [amountValue, splits])

```

```

function redistributeSplits(currentSplits, amount) { if (!Number.isFinite(amount) || amount < 0) amount = 0 const
selected = currentSplits.filter((split) => split.isSelected) const manualTotal = selected .filter((split) =>
split.isManual) .reduce((sum, split) => sum + split.amount, 0) const remaining = Math.max(amount - manualTotal,
0) const autoParticipants = selected.filter((split) => !split.isManual) const autoCount = autoParticipants.length

```

```

let autoShare = autoCount ? roundTwo(remaining / autoCount) : 0

let updatedSplits = currentSplits.map((split) => {
  if (!split.isSelected) {
    return { ...split, amount: 0, isManual: false }
  }
  if (split.isManual || autoCount === 0) return split
  return { ...split, amount: autoShare }
})

if (autoCount) {
  const autoIds = autoParticipants.map((split) => split.id)
  const autoTotal = autoShare * autoCount
  const diff = roundTwo(remaining - autoTotal)
  if (Math.abs(diff) >= 0.01) {
    const lastAutoId = autoIds[autoIds.length - 1]
    updatedSplits = updatedSplits.map((split) => {
      if (split.id === lastAutoId) {
        const adjusted = roundTwo(split.amount + diff)
        return { ...split, amount: adjusted < 0 ? 0 : adjusted }
      }
      return split
    })
  }
}

return updatedSplits

```

```

}

```

```

function handleAmountChange(e) { setAmountInput(e.target.value) }

```

```

function toggleParticipant(memberId) { setSplits((prev) => { const next = prev.map((split) => { if (split.id !==
memberId) return split const nextSelected = !split.isSelected return { ...split, isSelected: nextSelected, amount:
nextSelected ? split.amount : 0, isManual: nextSelected ? split.isManual : false, } }) return redistributeSplits(next,
amountValue) }) }

```

```
function handleManualAmount(memberId, value) { setSplits((prev) => { const amountCap =
Number.isFinite(amountValue) ? amountValue : Infinity const otherManualTotal = prev .filter((split) => split.id !==
memberId && split.isSelected && split.isManual) .reduce((sum, split) => sum + split.amount, 0)
```

```
const next = prev.map((split) => {
  if (split.id !== memberId) return split
  if (!split.isSelected) return split
  const parsed = parseFloat(value)
  const candidate = Number.isFinite(parsed) ? Math.max(parsed, 0) : 0
  const maxAllowed = amountCap === Infinity ? candidate : Math.max(amountCap
- otherManualTotal, 0)
  const clipped = roundTwo(Math.min(candidate, maxAllowed))
  return {
    ...split,
    amount: clipped,
    isManual: true,
  }
})
return redistributeSplits(next, amountValue)
})
```

```
}

function resetManual(memberId) { setSplits((prev) => { const next = prev.map((split) => { if (split.id !==
memberId) return split return { ...split, isManual: false, } }) return redistributeSplits(next, amountValue) }) }
```

```
function handleSubmit(e) { e.preventDefault() if (!groupId) return
```

```
const trimmedDescription = description.trim()
if (!trimmedDescription) {
  setError('Description is required.')
  return
}

if (!Number.isFinite(amountValue) || amountValue <= 0) {
  setError('Enter a valid amount.')
  return
}

const selectedSplits = splits.filter((split) => split.isSelected)
if (selectedSplits.length === 0) {
  setError('Select at least one participant.')
  return
}

const totalAllocated = roundTwo(selectedSplits.reduce((acc, split) => acc +
split.amount, 0))
if (Math.abs(totalAllocated - amountValue) > 0.01) {
  setError('Splits must add up to the total amount.')
  return
}
```



```
const payloadSplits = selectedSplits.map((split) => ({
  user: { id: split.id, name: split.name },
  amount: roundTwo(split.amount),
}))

const payloadTotal = roundTwo(payloadSplits.reduce((acc, split) => acc +
split.amount, 0))
const diff = roundTwo(amountValue - payloadTotal)
if (Math.abs(diff) > 0 && payloadSplits.length) {
  const lastIndex = payloadSplits.length - 1
  payloadSplits[lastIndex] = {
    ...payloadSplits[lastIndex],
    amount: roundTwo(payloadSplits[lastIndex].amount + diff),
  }
}

const payerId = paidById ?? selectedSplits[0].id

setIsSubmitting(true)

if (mode === 'edit' && expense) {
  const roundedAmount = roundTwo(amountValue)
  const expensePayload = {
    description: trimmedDescription,
    amount: roundedAmount,
    paid_by_id: payerId,
    photo_url: expense?.photo_url ?? null,
    splits: payloadSplits,
  }

  updateGroupExpense(groupId, {
    id: expense.id,
    expense: expensePayload,
  })
    .then((updatedExpense) => {
      onSuccess(updatedExpense, 'edit')
      onClose()
    })
    .catch((err) => {
      setError(err.message || 'Failed to update expense.')
    })
    .finally(() => setIsSubmitting(false))
  return
}

createGroupExpense(groupId, {
  description: trimmedDescription,
  amount: roundTwo(amountValue),
  paid_by_id: payerId,
  photo_url: null,
  splits: payloadSplits,
})
  .then((createdExpense) => {
```

```

    onSuccess(createdExpense, 'create')
    onClose()
  })
  .catch((err) => {
    setError(err.message || 'Failed to create expense.')
  })
  .finally(() => setIsSubmitting(false))

```

```

}

```

```

const headerLabel = mode === 'edit' ? 'Edit expense' : 'Add expense' const actionLabel = mode === 'edit' ? 'Edit expense' : 'Add expense'

```

```

return ( <Dialog open onOpenChange={(open) => { if (!open) onClose() }}> {headerLabel} Split an expense across members and keep balances up to date.

```

```

Description <Input id="expense-description" type="text" value={description} onChange={(e) => setDescription(e.target.value)} placeholder="What was the expense for?" required />

```

```

Amount 0.00

```

```

Paid by <Select value={paidById != null ? String(paidById) : undefined} onValueChange={(value) => setPaidById(Number(value))} > {members.map((member) => ( {member.name} ))}

```

```

Split between

```

```

{splits.map((split) => ( <div key={split.id} className={cn( 'flex flex-col gap-2 rounded-lg border border-border/80 bg-card/40 p-3 sm:flex-row sm:items-center sm:gap-3', !split.isSelected && 'opacity-60' )} >
<Checkbox id={split-$split.id} checked={split.isSelected} onCheckedChange={() => toggleParticipant(split.id)} disabled={isSubmitting} /> <Label htmlFor={split-$split.id} className="flex-1 text-base font-medium"> {split.name} {split.isManual && split.isSelected && ( <Button type="button" size="sm" variant="ghost" onClick={() => resetManual(split.id)} disabled={isSubmitting} > Reset )}
<Input type="number" step="0.01" min="0" value={split.isSelected ? split.amount : 0} onChange={(e) => handleManualAmount(split.id, e.target.value)} disabled={!split.isSelected || isSubmitting} />
)))
{error && (

```

```

{error}

```

```

)) <Cancel <{actionLabel}
))

```

```

===== FILE: frontend/src/features/groups/GroupDetails.jsx ===== import { useEffect, useMemo, useRef, useState } from 'react' import { useLocation, useParams } from 'react-router-dom' import { createGroupInviteLink, deleteGroupExpense, getGroupDetails } from '../services/api.js' import { getCachedGroupDetailsByName, getGroupIdByName, saveGroupDetails } from './groupStorage.js' import AddExpenseModal from './AddExpenseModal.jsx' import { useAuth } from '../AuthContext.jsx' import { Button } from '../components/ui/button.jsx' import { Tabs, TabsList, TabsTrigger, TabsContent } from '../components/ui/tabs.jsx' import { Badge } from '../components/ui/badge.jsx' import { Card } from '../components/ui/card.jsx' import { Dialog, DialogContent, DialogDescription, DialogFooter, DialogHeader, DialogTitle, } from '../components/ui/dialog.jsx' import { Input } from '../components/ui/input.jsx' import { cn } from '../lib/utlis.js' import { Copy, Pencil, Plus, ReceiptText, Trash2 } from 'lucide-react'

```


```

function formatCurrency(amount) { if (typeof amount !== 'number') return '—' return new Intl.NumberFormat(undefined, { style: 'currency', currency: 'USD' }).format(amount) }

```

```
function ExpenseRow({ expense, expanded, onToggle, onEdit, onDelete }) { const title = expense.description ||
'Untitled expense' const createdByName = expense.created_by?.name || expense.paid_by?.name || 'someone'

return ( <Card role="button" tabIndex={0} onClick={onToggle} onKeyDown={(event) => { if (event.key === 'Enter'
|| event.key === ' ') { event.preventDefault() onToggle() } }} className={cn( 'cursor-pointer select-none transition
hover:border-primary/60 focus:outline-none focus-visible:ring-2 focus-visible:ring-ring', expanded && 'border-
primary/60' )} >
```

```
{expense.photo_url ? (  {title} ) : ( )}
```

```
{title}
```

```
Created by {createdByName} {formatCurrency(typeof expense.amount === 'number' ? expense.amount :
Number(expense.amount) || 0)}
```

```
<Button type="button" variant="ghost" size="icon" onClick={(event) => { event.stopPropagation()
onEdit(expense) }} > <Button type="button" variant="ghost" size="icon" className="text-destructive hover:text-
destructive" onClick={(event) => { event.stopPropagation() onDelete(expense) }} >
```

```
{expanded ? (
```

```
Paid by {expense.paid_by?.name || 'Unknown'}
```

```
Splits
```

```
{Array.isArray(expense.splits) && expense.splits.length ? (
{expense.splits.map((split) => { const splitName = split.user?.name || 'Member' const splitAmount = typeof
split.amount === 'number' ? split.amount : Number(split.amount) || 0 return ( <div key=
`${expense.id}-${split.user?.id ?? splitName}` className="flex items-center justify-between text-
sm"> {splitName} {formatCurrency(splitAmount)}
)}}}
): (
```

```
No split information available.
```

```
)}
```

```
): null}
```

```
)}
```

```
function BalanceRow({ member, isCurrentUser }) { const balanceValue = typeof member.balance === 'number' ?
member.balance : Number(member.balance) || 0 const amountDisplayBase =
formatCurrency(Math.abs(balanceValue)) const amountDisplay = balanceValue > 0 ? `${amountDisplayBase}` :
balanceValue < 0 ? `${-amountDisplayBase}` : amountDisplayBase const amountClasses = balanceValue > 0 ?
'text-emerald-300' : balanceValue < 0 ? 'text-rose-300' : 'text-muted-foreground'
```

```
return (
```

```
{member.name}
```

```
{isCurrentUser ? Me : null}
```

```
<span className={cn('text-lg font-semibold', amountClasses)}>{amountDisplay} ) }
```

```
export default function GroupDetails() { const { groupName } = useParams() const decodedName = useMemo(()
=> decodeURIComponent(groupName), [groupName]) const location = useLocation() const { userId } = useAuth()
```

```
const initialGroup = location.state?.group || getCacheGroupDetailsByName(decodedName) const [group,
setGroup] = useState(initialGroup || null) const [isLoading, setIsLoading] = useState(!initialGroup) const [error,
setError] = useState(null) const [activeTab, setActiveTab] = useState('expenses') const [modalState,
```

```

setModalState] = useState(null) const [expandedExpenseId, setExpandedExpenseId] = useState(null) const
[isInviteLoading, setIsInviteLoading] = useState(false) const [inviteDialogOpen, setInviteDialogOpen] =
useState(false) const [inviteUrl, setInviteUrl] = useState("") const [copySuccess, setCopySuccess] =
useState(false) const [inviteError, setInviteError] = useState(null) const [copyError, setCopyError] = useState(null)
const copyTimeoutRef = useRef(null) const memberMapKey = useMemo(() =>
memberNameToId: ${decodedName}, [decodedName])

```

```

const memberNameTold = useMemo(() => { if (!group?.members?.length) return {} return
group.members.reduce((acc, member) => { if (member?.name && member?.id != null) { acc[member.name] =
member.id } return acc }, {}), [group])

```

```

useEffect(() => { return () => { if (copyTimeoutRef.current) { clearTimeout(copyTimeoutRef.current) } }, [])

```

```

useEffect(() => { if (Object.keys(memberNameTold).length) { try { sessionStorage.setItem(memberMapKey,
JSON.stringify(memberNameTold)) } catch (err) { console.error('Failed to cache member name map', err) } } else {
try { sessionStorage.removeItem(memberMapKey) } catch (err) { console.error('Failed to clear member name
map', err) } } }

```

```

return () => {
  try {
    sessionStorage.removeItem(memberMapKey)
  } catch (err) {
    console.error('Failed to remove member name map', err)
  }
}

```

```

}, [memberMapKey, memberNameTold])

```

```

const groupId = useMemo(() => { if (location.state?.groupId) return location.state.groupId if (initialGroup?.id)
return initialGroup.id return getGroupIdByName(decodedName) }, [location.state, initialGroup, decodedName])

```

```

const numericUserId = useMemo(() => { if (!userId) return null const parsed = parseInt(userId, 10) return
Number.isFinite(parsed) ? parsed : null }, [userId])

```

```

const totals = useMemo(() => { const expenses = Array.isArray(group?.expenses) ? group.expenses : [] const
totalExpenses = expenses.reduce((sum, expense) => { const amount = typeof expense.amount === 'number' ?
expense.amount : Number(expense.amount) || 0 return sum + amount }, 0)

```

```

const myExpenses = expenses.reduce((sum, expense) => {
  if (!numericUserId || !Array.isArray(expense.splits)) return sum
  const matchingSplit = expense.splits.find((split) => {
    const splitId = split.user?.id ?? split.user_id
    return splitId === numericUserId
  })
  if (!matchingSplit) return sum
  const splitAmount = typeof matchingSplit.amount === 'number' ?
matchingSplit.amount : Number(matchingSplit.amount) || 0
  return sum + splitAmount
}, 0)

return {

```

```

    totalExpenses,
    myExpenses,
  }

```

```

}, [group?.expenses, numericUserId])

```

```

const currentMemberBalance = useMemo(() => { if (!numericUserId || !Array.isArray(group?.members)) return null
const me = group.members.find((member) => member.id === numericUserId) if (!me) return null const
balanceValue = typeof me.balance === 'number' ? me.balance : Number(me.balance) || 0 return balanceValue },
[group?.members, numericUserId])

```

```

const otherMemberCount = useMemo(() => { if (!Array.isArray(group?.members)) return 0 if (!numericUserId)
return group.members.length return group.members.filter((member) => member.id !== numericUserId).length },
[group?.members, numericUserId])

```

```

const isSoloGroup = otherMemberCount === 0 const inviteButtonLabel = isInviteLoading ? 'Generating...' : 'Invite a
Friend!'

```

```

useEffect(() => { setCopySuccess(false) setCopyError(null) if (copyTimeoutRef.current) {
clearTimeout(copyTimeoutRef.current) copyTimeoutRef.current = null } }, [inviteUrl])

```

```

useEffect(() => { let isMounted = true

```

```

async function loadDetails() {
  if (group) {
    setIsLoading(false)
    setError(null)
    return
  }
  if (!groupId) {
    setIsLoading(false)
    setError('Group details not found. Try reopening from your account page.')
    return
  }

  setIsLoading(true)
  setError(null)

  try {
    const data = await getGroupDetails(groupId)
    if (!isMounted) return
    setGroup(data)
    saveGroupDetails(data)
  } catch (err) {
    if (!isMounted) return
    setError(err.message || 'Failed to load group details')
  } finally {
    if (isMounted) setIsLoading(false)
  }
}

loadDetails()

```

```

return () => {
  isMounted = false
}

```

```

}, [group, groupId])

```

```

async function refreshGroupDetails() { if (!groupId) return try { const data = await getGroupDetails(groupId)
setGroup(data) saveGroupDetails(data) } catch (err) { console.error('Failed to refresh group after creating
expense', err) } }

```

```

function handleExpenseSuccess(expenseResult, type) { setGroup((prev) => { if (!prev) return prev const
existingExpenses = Array.isArray(prev.expenses) ? prev.expenses : [] if (type === 'edit') { return { ...prev,
expenses: existingExpenses.map((expense) => (expense.id === expenseResult.id ? expenseResult : expense)), } }
return { ...prev, expenses: [expenseResult, ...existingExpenses], } }) refreshGroupDetails() }

```

```

function handleOpenCreateModal() { setModalState({ mode: 'create' }) }

```

```

function handleOpenEditModal(expense) { setModalState({ mode: 'edit', expense }) }

```

```

function handleCloseModal() { setModalState(null) }

```

```

async function handleDeleteExpense(expense) { if (!groupId || !expense?.id) return try { await
deleteGroupExpense(groupId, { id: expense.id }) setGroup((prev) => { if (!prev) return prev const existingExpenses
= Array.isArray(prev.expenses) ? prev.expenses : [] return { ...prev, expenses: existingExpenses.filter((item) =>
item.id !== expense.id), } }) refreshGroupDetails() } catch (err) { console.error('Failed to delete expense', err) } }

```

```

function handleToggleExpense(expenseId) { setExpandedExpenseId((prev) => (prev === expenseId ? null :
expenseId)) }

```

```

async function handleInviteClick() { if (!groupId) return setIsInviteLoading(true) try { setInviteError(null)
setCopyError(null) setCopySuccess(false) const data = await createGroupInviteLink(groupId) const token =
data?.token if (!token) { throw new Error('Invite token was not returned') } const shareUrl =
`${window.location.origin}/join?token=${encodeURIComponent(token)} setInviteUrl(shareUrl)
setInviteDialogOpen(true) } catch (err) { console.error('Failed to create invite link', err) const message = err
instanceof Error ? err.message : 'Failed to create invite link' setInviteError(message) } finally {
setIsInviteLoading(false) } }

```

```

async function handleCopyInvite() { if (!inviteUrl) return try { setCopyError(null) await
navigator.clipboard.writeText(inviteUrl) setCopySuccess(true) if (copyTimeoutRef.current) {
clearTimeout(copyTimeoutRef.current) } copyTimeoutRef.current = setTimeout(() => { setCopySuccess(false) },
2000) } catch (err) { console.error('Failed to copy invite link', err) const message = err instanceof Error ?
err.message : 'Unable to copy link. Please copy it manually.' setCopyError(message) setCopySuccess(false) } }

```

```

if (isLoading) { return (

```

```

  Loading group details...

```

```

) }

```

```

if (error) { return (

```

```

  {error}

```

```

) }

```

```
if (!group) { return (
```

```
  Group data is unavailable.
```

```
  )}
```

```
return (
```

```
{group.emoji || '👥'}
```

```
{group.name}
```

```
{group.members?.length || 0} members
```

```
{inviteButtonLabel}
```

```
Add expense
```

```
{inviteError ?
```

```
{inviteError}
```

```
: null}
```

```
<div className="grid gap-3 sm:grid-cols-2">
  <Card className="p-4">
    <p className="text-sm text-muted-foreground">My expenses</p>
    <p className="text-2xl font-semibold">{formatCurrency(totals.myExpenses)}</p>
  </Card>
  <Card className="p-4">
    <p className="text-sm text-muted-foreground">Total expenses</p>
    <p className="text-2xl font-semibold">{formatCurrency(totals.totalExpenses)}</p>
  </Card>
</div>

<Tabs value={activeTab} onValueChange={setActiveTab} className="space-y-4">
  <TabsList>
    <TabsTrigger value="expenses">Expenses</TabsTrigger>
    <TabsTrigger value="balances">Balances</TabsTrigger>
  </TabsList>

  <TabsContent value="expenses" className="space-y-4">
    {group.expenses?.length ? (
      <div className="space-y-3">
        {group.expenses.map((expense) => (
          <ExpenseRow
            key={expense.id}
            expense={expense}
            expanded={expandedExpenseId === expense.id}
            onToggle={() => handleToggleExpense(expense.id)}
            onEdit={handleOpenEditModal}
            onDelete={handleDeleteExpense}
          />
        ))}
      </div>
    ) : (
```

```

    <div className="rounded-lg border border-dashed border-muted/50 bg-
muted/20 p-8 text-center text-sm text-muted-foreground">
      No expenses recorded yet.
    </div>
  )}
</TabsContent>

<TabsContent value="balances" className="space-y-4">
  {isSoloGroup ? (
    <Card className="flex flex-col items-center gap-3 p-6 text-center">
      <p className="text-lg font-medium">Invite friends to start sharing
expenses.</p>
      <Button
        type="button"
        variant="outline"
        onClick={handleInviteClick}
        disabled={isInviteLoading}
      >
        {inviteButtonLabel}
      </Button>
      {inviteError ? <p className="text-sm text-destructive">{inviteError}
</p> : null}
    </Card>
  ) : (
    <>
      {typeof currentMemberBalance === 'number' ? (
        <Card className="p-5">
          {currentMemberBalance > 0 ? (
            <p className="text-xl font-semibold text-emerald-300">
              You are owed {formatCurrency(Math.abs(currentMemberBalance))}
            </p>
          ) : currentMemberBalance < 0 ? (
            <p className="text-xl font-semibold text-rose-300">
              You owe {formatCurrency(Math.abs(currentMemberBalance))}
            </p>
          ) : (
            <p className="text-xl font-semibold text-muted-foreground">You
are all settled</p>
          )}
        </Card>
      ) : null}

      {group.members?.length ? (
        <div className="space-y-3">
          {group.members.map((member) => (
            <BalanceRow key={member.id} member={member} isCurrentUser=
{numericUserId === member.id} />
          ))}
        </div>
      ) : (
        <div className="rounded-lg border border-dashed border-muted/50 bg-
muted/20 p-8 text-center text-sm text-muted-foreground">
          No balances to show.
        </div>
      )}
    </>
  )}

```



```

    })
  </>
})
</TabsContent>
</Tabs>

<Dialog
  open={inviteDialogOpen}
  onOpenChange={({open) => {
    setInviteDialogOpen(open)
    if (!open) {
      setCopySuccess(false)
      setCopyError(null)
      if (copyTimeoutRef.current) {
        clearTimeout(copyTimeoutRef.current)
        copyTimeoutRef.current = null
      }
    }
  }}
  >
  <DialogContent>
    <DialogHeader>
      <DialogTitle>Share your invite link</DialogTitle>
      <DialogDescription>Send this link to friends so they can join your
group.</DialogDescription>
    </DialogHeader>
    <div className="space-y-2">
      <p className="text-sm text-muted-foreground">Invite link</p>
      <Input
        value={inviteUrl}
        readOnly
        onFocus={(event) => event.target.select()}
      />
    </div>
    {copyError ? <p className="text-sm text-destructive">{copyError}</p> :
null}
    <DialogFooter className="flex flex-col gap-2 sm:flex-row sm:justify-end">
      <Button type="button" variant="outline" onClick={() =>
setInviteDialogOpen(false)}>
        Close
      </Button>
      <Button type="button" onClick={handleCopyInvite} disabled={!inviteUrl}>
        <Copy className="mr-2 h-4 w-4" /> {copySuccess ? 'Copied!' : 'Copy
link'}
      </Button>
    </DialogFooter>
  </DialogContent>
</Dialog>

{modalState && group?.members?.length ? (
  <AddExpenseModal
    groupId={group.id}
    members={group.members}
    currentUserId={numericUserId}

```

```

      onClose={handleCloseModal}
      onSuccess={handleExpenseSuccess}
      mode={modalState.mode}
      expense={modalState.expense}
    />
  ) : null}
</section>

```

```

  )}

===== FILE: frontend/src/features/groups/groupStorage.js =====
const GROUP_DETAILS_KEY =
'groupDetailsById'
const GROUP_NAME_MAP_KEY = 'groupNameTold'

function safeRead(key) {
  try {
    const raw = sessionStorage.getItem(key)
    return raw ? JSON.parse(raw) : {}
  } catch (err) {
    console.error('Failed to read storage key', key, err)
    return {}
  }
}

function safeWrite(key, value) {
  try {
    sessionStorage.setItem(key, JSON.stringify(value))
  } catch (err) {
    console.error('Failed to write storage key', key, err)
  }
}

export function rememberGroupId(name, id) {
  if (!name || !id) return
  const map = safeRead(GROUP_NAME_MAP_KEY)
  map[name] = id
  safeWrite(GROUP_NAME_MAP_KEY, map)
}

export function getGroupIdByName(name) {
  if (!name) return null
  const map = safeRead(GROUP_NAME_MAP_KEY)
  return map[name] ?? null
}

export function saveGroupDetails(group) {
  if (!group?.id) return
  const cache = safeRead(GROUP_DETAILS_KEY)
  cache[group.id] = group
  safeWrite(GROUP_DETAILS_KEY, cache)
  rememberGroupId(group.name, group.id)
}

export function getCachedGroupDetailsById(id) {
  if (!id) return null
  const cache = safeRead(GROUP_DETAILS_KEY)
  return cache[id] ?? null
}

export function getCachedGroupDetailsByName(name) {
  const groupId = getGroupIdByName(name)
  if (!groupId) return null
  return getCachedGroupDetailsById(groupId)
}

export function syncGroupSummaries(groups) {
  if (!Array.isArray(groups)) return
  const map = safeRead(GROUP_NAME_MAP_KEY)
  for (const group of groups) {
    if (group?.name && group?.id) {
      map[group.name] = group.id
    }
  }
  safeWrite(GROUP_NAME_MAP_KEY, map)
}

===== FILE: frontend/src/index.css =====
@tailwind base;
@tailwind components;
@tailwind utilities;

@layer base {
  :root {
    --background: 221 39% 7%;
    --foreground: 220 27% 90%;
    --card: 221 39% 9%;
    --card-foreground: 220 27% 90%;
    --popover: 221 39% 9%;
    --popover-foreground: 220 27% 90%;
    --primary: 220 100% 60%;
    --primary-foreground: 210 40% 98%;
    --secondary: 218 25% 22%;
    --secondary-foreground: 220 27% 90%;
    --muted: 220 23% 18%;
    --muted-foreground: 215 20% 65%;
    --accent: 215 63% 48%;
    --accent-foreground: 210 40% 98%;
    --destructive: 0 72% 51%;
    --destructive-foreground: 210 40% 98%;
    --border: 214 23% 24%;
    --input: 214 23% 24%;
    --ring: 217 91% 60%;
    --radius: 0.75rem;
  }

  • { @apply border-border; }

  body { @apply bg-background text-foreground antialiased; min-height: 100vh; }

  button { @apply font-medium; }

  a { @apply text-primary underline-offset-4 transition-colors hover:underline; }

```

```

===== FILE: frontend/src/lib/utlis.js ===== import { clsx } from "clsx" import { twMerge } from "tailwind-merge"

export function cn(...inputs) { return twMerge(clsx(inputs)) }

===== FILE: frontend/src/main.jsx ===== import { createRoot } from 'react-dom/client' import { BrowserRouter }
from 'react-router-dom' import './index.css' import App from './App.jsx' import { AuthProvider } from
'./AuthContext.jsx'

createRoot(document.getElementById('root')).render( )

===== FILE: frontend/src/services/api.js ===== const API_BASE_URL = import.meta.env.VITE_API_BASE_URL ||
"http://127.0.0.1:8000";

function getStoredToken() { return localStorage.getItem("token"); }

function getAuthHeaders(extraHeaders = {}) { const token = getStoredToken(); const headers = { "Content-Type":
"application/json", ...extraHeaders }; if (token) headers["Authorization"] = `Bearer ${token}`; return headers; }

async function handleResponse(res) { if (res.status === 401) { localStorage.removeItem("token");
localStorage.removeItem("userId"); window.location.replace("/login"); throw new Error("Unauthorized"); } if
(!res.ok) { let message = "Request failed"; const errorPayload = await res.clone().json().catch(() => null); if
(errorPayload?.detail || errorPayload?.message) { message = errorPayload.detail || errorPayload.message ||
message; } throw new Error(message); } const data = await res.json().catch(() => null); return data; }

export async function apiGet(path) { const res = await fetch( `${API_BASE_URL}${path}`, { method: "GET",
headers: getAuthHeaders(), credentials: "include", }); return handleResponse(res); }

export async function apiPost(path, body) { const res = await fetch( `${API_BASE_URL}${path}`, { method:
"POST", headers: getAuthHeaders(), body: JSON.stringify(body), credentials: "include", }); return
handleResponse(res); }

// High-level helpers (adjust to your backend as needed) export async function getMessage() { return apiGet("/"); }

// Signup: expects backend to return { access_token, user: { id, name, email } } export async function
signupUser(payload) { const data = await apiPost("/auth/signup", payload); if (data?.access_token)
localStorage.setItem("token", data.access_token); if (data?.user?.id) localStorage.setItem("userId",
String(data.user.id)); return data; }

// Login: expects backend to return { access_token, user: { id, name, email } } export async function
loginUser(payload) { const data = await apiPost("/auth/login", payload); if (data?.access_token)
localStorage.setItem("token", data.access_token); if (data?.user?.id) localStorage.setItem("userId",
String(data.user.id)); return data; }

export async function createGroup(payload) { return apiPost("/groups/create", payload); }

export async function joinGroup(payload) { return apiPost("/groups/join", payload); }

export async function getUserGroups() { return apiGet("/groups/view-short"); }

export async function getGroupDetails(groupId) { return apiGet( `/groups/${groupId}`); }

export async function createGroupInviteLink(groupId) { return apiGet( `/groups/${groupId}/create-invite`);
}

```

```

export async function createGroupExpense(groupId, payload) { return
apiPost(/expenses/${groupId}/create-expense, payload); }

export async function updateGroupExpense(groupId, payload) { return apiPost( /expenses/${groupId}/edit-
expense, payload); }

export async function deleteGroupExpense(groupId, payload) { return
apiPost(/expenses/${groupId}/delete-expense, payload); }

export function logout() { localStorage.removeItem("token"); localStorage.removeItem("userId");
window.location.replace(""); }

===== FILE: frontend/tailwind.config.js ===== import tailwindcssAnimate from "tailwindcss-animate"

/** @type {import('tailwindcss').Config} / export default { darkMode: ["class"], content: [ "./index.html", "./src/**/
{.js,.jsx,.ts,.tsx}" ], theme: { container: { center: true, padding: "2rem", screens: { "2xl": "1400px" } }, extend: { colors:
{ border: "hsl(var(--border))", input: "hsl(var(--input))", ring: "hsl(var(--ring))", background: "hsl(var(--
background))", foreground: "hsl(var(--foreground))", primary: { DEFAULT: "hsl(var(--primary))", foreground:
"hsl(var(--primary-foreground))" }, secondary: { DEFAULT: "hsl(var(--secondary))", foreground: "hsl(var(--
secondary-foreground))" }, destructive: { DEFAULT: "hsl(var(--destructive))", foreground: "hsl(var(--destructive-
foreground))" }, muted: { DEFAULT: "hsl(var(--muted))", foreground: "hsl(var(--muted-foreground))" }, accent: {
DEFAULT: "hsl(var(--accent))", foreground: "hsl(var(--accent-foreground))" }, popover: { DEFAULT: "hsl(var(--
popover))", foreground: "hsl(var(--popover-foreground))" }, card: { DEFAULT: "hsl(var(--card))", foreground:
"hsl(var(--card-foreground))" } }, borderRadius: { lg: "var(--radius)", md: "calc(var(--radius) - 2px)", sm:
"calc(var(--radius) - 4px)" }, keyframes: { "accordion-down": { from: { height: 0 }, to: { height: "var(--radix-
accordion-content-height)" } }, "accordion-up": { from: { height: "var(--radix-accordion-content-height)" }, to: {
height: 0 } } }, animation: { "accordion-down": "accordion-down 0.2s ease-out", "accordion-up": "accordion-up
0.2s ease-out" } } }, plugins: [tailwindcssAnimate] }

===== FILE: frontend/vite.config.js ===== import { defineConfig } from 'vite' import react from '@vitejs/plugin-
react'

// https://vite.dev/config/ export default defineConfig({ plugins: [react()], })

```