Group-3 | CSE299.4
Safat Uddin (2311966642)
Farhan Karim (2311790642)
Faizur Rahman Zunayed (2312137642)

Backend (Done by Safat):

- Set up Express server with CORS and request logging (morgan), environment variables via dotenv, and MongoDB connection using Mongoose.
- Implemented auth features with validation using zod/zod-express-middleware:
    - Register: creates a user with hashed password (bcrypt), prevents duplicate emails, and sends a verification email via SendGrid with a JWT token stored in a Verification collection.
    - Verify Email: verifies the token, marks the user as verified, deletes the verification record.
    - Login: validates credentials, enforces email verification, regenerates verification if needed, returns a 7-day JWT and user data (password omitted), updates last login.
    - Forgot/Reset Password: creates and emails a time-limited JWT link; token is checked before allowing password reset.
- Security and safety: JWT-based flows, CORS tied to FRONTEND_URL, and Arcjet email inspection to prevent abuse.

Frontend (Done by Farhan & Zunayed):

- React Router v7 setup with TypeScript, Tailwind CSS, and component library wiring. App includes forms with react-hook-form + zod validation and toast feedback.
- Auth pages and flows implemented:
    - The Home page provides an introduction to the app, with clear calls-to-action for users to either sign up or sign in. **(Done by Farhan)**
    - Sign Up with client-side validation and success redirect to Sign In. **(Done by Zunayed)**
    - Sign In that stores auth state via a provider and navigates to Dashboard. **(Done by Zunayed)**
    - Email verification screen that reads the token from the URL and calls the backend. **(Done by Farhan)**
    - Forgot Password and Reset Password screens that orchestrate the email and token-based reset process. **(Done by Farhan)**
- Project structure with routes, hooks for API calls, and basic dashboard/home route scaffolding.

Project Flow (how it works end-to-end):

- A new user signs up on the frontend; the backend creates the user, generates a verification JWT, stores it, and emails a verification link that points back to the frontend.
- The user clicks the link; the frontend verifies the token with the backend. After success, the user can log in.
- On login, the backend validates password and verification status. If unverified/expired, a new verification flow is triggered. On success, a login JWT is returned and the frontend stores session state.
- If the user forgets their password, the frontend starts a reset flow; the backend generates a time-limited reset token, emails a link, and on confirmation updates the password.