**ProDraftfy**

**Smart Email Generator for Students**

* **High Level Design (HLD)**

Generate Email / Check Grammer

**Generate Email**

React UI

(Prompt Input + Options)

**Check Email Format and Grammer**

Submit Button

React UI

Check Grammar and Formatting

Flask API: Receives Request

Is Email Format Valid?

Call Gemini API for Email Generation

Show "Email is Valid and Grammatically Correct"

Display Draft on UI

Show "Invalid Email Format" Message

Export Option

Show Grammar Errors and Suggestions

Copy to Clipboard

Download as PDF

**Low-Level Design (LLD)**

* **Frontend (React.js)**
* **App.js**
  + I’m using **React Router** for navigation between the main screens.
  + The homepage acts as a landing interface where users can select:
    - Generate Email
    - Check Grammar & Format

<Routes>

<Route path="/" element={<Home />} />

<Route path="/generate" element={<EmailGenerator />} />

<Route path="/check" element={<GrammarCheck />} />

</Routes>

* **Home Component**
* This is a simple card-style UI that gives users a clear choice on what they want to do.
* I’ve styled the buttons with fixed dimensions and added purple as the primary color for branding consistency.
* **Features:**
* Prompt title and subtext
* Two CTA buttons leading to generation and grammar-checking flows
* **Pages**
* **EmailGenerator.js**
  + Handles the email generation from user prompt.
  + Inputs include:
    - **Prompt Text**
      * **Category** (dropdown: Leave, Complaint, etc.)
      * **Tone** (dropdown: Formal, Polite)
      * **Etc..**
* After submission:
  + The data is sent to the backend (/generate-email)
  + Response includes a well-structured and grammatically correct email
* **2. GrammarCheck.js**
  + This page accepts a custom user-written email.
  + Once submitted, the text is sent to the backend (/check-grammar)
  + The corrected version is returned and shown to the user with options to edit or export.
* **Design Elements**
* **Blobs** in the background are animated using CSS and give a unique and modern look to the interface.
* Centralized container with max-width for better responsiveness.
* Buttons are styled consistently with rounded corners and flex properties to align icons and text.