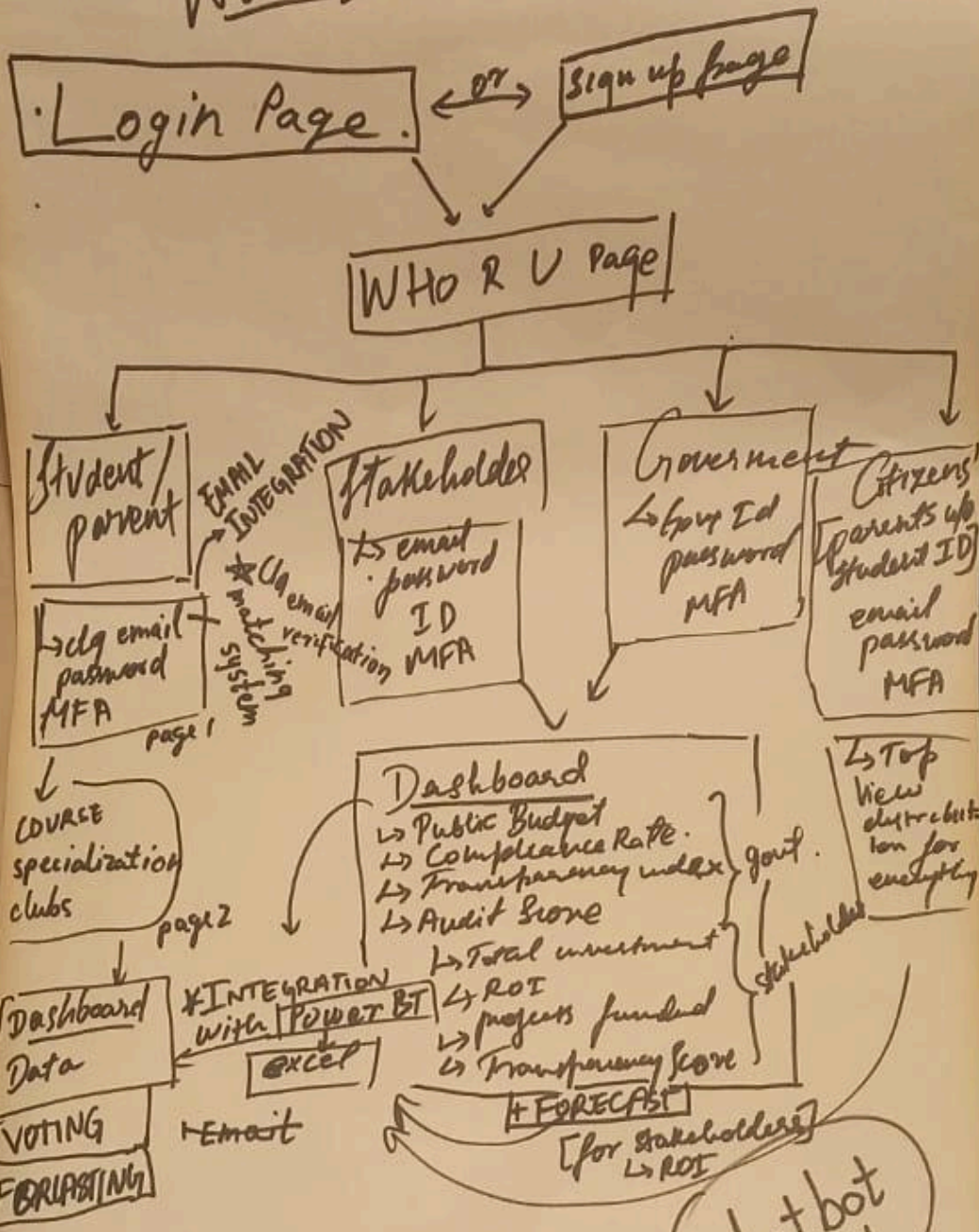


WEB INFRASTRUCTURE



Using Canva AI to create a base framework for html/ However processing code in VS code with react will take too long. So, shifting to an automated website creating software wix, and linking it to one of the cloud computing softwares for live updates.

WEB INFRASTRUCTURE

Overall Style:

Theme: Modern, professional, futuristic, and trustworthy.

Color Palette: Blue + White gradients (primary), soft grays for backgrounds, occasional green for “success” messages.

Typography: Clean sans-serif fonts, large bold titles, readable body text.

Elements: Rounded edges, subtle drop shadows, icons for categories, and clean layouts.

User experience should feel secure, transparent, and engaging.

Page 1: Login Page

Header: Company name Fiscality at the top, large bold font, centered.

Login box (centered card with shadow):

Username input field.

Password input field (hidden text).

‘Done’ button (rounded, primary blue).

If username/password fields have < 5 characters → show a popup warning (“Minimum character length required”).

Additional options (below box):

Sign Up link → Leads to a Sign Up Page.

Forgot Password link → Opens a small box:

Field: Enter your registered email.

On submit: “Reset email sent successfully.”

Popup message confirms → redirects back to Login page.

Sign Up Page

Title: "Create an Account"

Fields:

Username

Passcode

Confirm Passcode

Email

'Sign Up' button → registers and redirects to Login page.

Page 2: Who Are You Page

Title (centered, bold): Who Are You?

Three large clickable selection cards with icons:

Student/Parent

Stakeholder

Government

Done button → Leads to Page 3 (with different flows depending on choice).

Page 3: Verification Pages

For Students/Parents:

Title: Verification Required

Fields:

College Email

Password

2-Step Authorization (a randomly generated code is emailed to the user's Gmail).

If code ≠ generated one → show "Invalid code".

Done button → Leads to Page 4 (Student Information).

For Stakeholders:

Title: Verification Required

Fields:

Email

Password

Stakeholder ID

2-Step Authorization (random code sent to Gmail).

If code \neq generated one \rightarrow show "Invalid code".

Done button \rightarrow Leads to Stakeholder Dashboard (Page 5).

For Government Officials:

Fields:

Government ID

Password (must be > 9 characters, otherwise show "Invalid password length").

Done button \rightarrow Leads directly to Government Dashboard (Page 5).

Page 4: Student Information Page

Title: Student Profile Setup

Input fields (dropdowns):

Course (list all courses from RVCE — Engineering, Architecture, MCA, MBA, etc.).

Specialization (Computer Science, Electronics, Mechanical, Civil, AI/ML, etc.).

Clubs (multi-select dropdown with RVCE clubs — music, debate, robotics, cultural, sports, etc.).

Done button \rightarrow Leads to Personalized Student Dashboard.

Page 5: Personalized Dashboard (Dynamic by User Type)

Shared Features (all types):

Connected Power BI visualizations (show sample charts & stats).

Personalization based on login type & details.

Search bar:

For students → Search only within their department, events, and clubs.

For stakeholders → Search projects, funds, and transparency reports.

For government → Search compliance, public data, and overall budgets.

Student Dashboard:

Displays financial transparency for their course, specialization, and clubs.

Key metrics: club budgets, department allocations, funding progress.

Stakeholder Dashboard:

Displays overview of investments, funds allocation, and transparency stats.

Emphasis on overall viewpoint and major financial indicators.

Government Dashboard:

Displays macro-level financial data, compliance stats, and public transparency reports.

Focus on governance and accountability.

Page 6: Voting & Forecast Page

Section title: Have Your Say.

Options to vote on:

Increase in funding.

Additional funding for clubs/departments.

Public petitions (with vote counters showing live tallies).

Note: "Only valid once enough votes are collected."

Vote Now button → increments live count.

Forecast Section:

Title: Financial Forecasting

Shows budget predictions & trends using past data.

Displays approximate future distributions (charts + graphs from Power BI).

okay for the gmail verification thing, just let the user enter anything and still go to the next page.

+ there should be only relative (and appropriate) voting if the user is not a student. like govt and stakeholders dont need it.

The image shows a VS Code editor window with a dark theme. The Explorer sidebar on the left shows a file named 'indexfinald.html'. The main editor area displays HTML and JavaScript code for a feedback form. The HTML includes a form with fields for feedback ID, category, priority, and status. The JavaScript code includes a 'showpopup()' function and a 'closefeedbackModal()' function. The code is partially obscured by an Excel spreadsheet titled 'College Expenditure Budget' which is in 'Protected View'.

The Excel spreadsheet has a green header bar with the title 'College Expenditure Budget [Protected View] - Excel'. Below the header bar, there is a ribbon with tabs for 'File', 'Home', 'Insert', 'Draw', 'Page Layout', 'Formulas', 'Data', 'Review', 'View', 'Help', and 'Tell me what you want to do'. A warning bar below the ribbon states: 'PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. Enable'. The spreadsheet itself has a grid with columns A through L and rows 541 through 564. The data in the spreadsheet is as follows:

	A	B	C	D	E	F	G	H	I	J	K	L
541	Medical	Sports	Dec	1470438								
542	Medical	Admin	Jan	414961								
543	Medical	Admin	Feb	1340828								
544	Medical	Admin	Mar	1243500								
545	Medical	Admin	Apr	681137								
546	Medical	Admin	May	901601								
547	Medical	Admin	Jun	1199107								
548	Medical	Admin	Jul	50851								
549	Medical	Admin	Aug	687271								
550	Medical	Admin	Sep	651701								
551	Medical	Admin	Oct	490785								
552	Medical	Admin	Nov	951540								
553	Medical	Admin	Dec	1266316								
554	Medical	Student W	Jan	1322261								
555	Medical	Student W	Feb	1423017								
556	Medical	Student W	Mar	725805								
557	Medical	Student W	Apr	1025150								
558	Medical	Student W	May	976492								
559	Medical	Student W	Jun	977276								
560	Medical	Student W	Jul	1466893								
561	Medical	Student W	Aug	498164								
562	Medical	Student W	Sep	1122009								
563	Medical	Student W	Oct	1042343								
564	Medical	Student W	Nov	405868								

The bottom of the Excel window shows the 'Sheet1' tab and a status bar with the word 'Ready'.

notes - Replit

https://replit.com/@kittenmanani/notes

notes + Upgrade

Add initial project setup for the Replit envi... Agent 3

Completed tasks 8 / 8

- Ask for Google API key secret ✓
- Copy HTML file to main directory ✓
- Add Firebase imports to HTML head ✓
- Create Firebase login, signup functions ✓
- Connect Firebase to forms ✓
- Embed Looker Studio in dashboard ✓
- Serve HTML file on port 5000 ✓
- Test Firebase sign-in functionality ✓

Add detailed documentation for the financial transparency platform

Rollback here <> Changes Preview

4 minutes of work

Make, test, iterate...

Build

Preview

Auth

Git

main

Remote Updates

origin/main • upstream

last fetched 33 min ago Fetch

6 commits to push

Sync with Remote Pull Push

Commit

There are no changes to commit.

Not pushed to remote

- Add detailed documentation for the financial transparency platform
 - kittenmanani just now
- Add Firebase authentication and Looker Studio integration to the application
 - kittenmanani 2 minutes ago
- Add styling to highlight selected items in the navigation bar
 - kittenmanani 13 minutes ago
- Add user authentication and account management features

Firestore

Docs More

Search /

English

Blog Studio Go to console

Overview Fundamentals AI Build Run Reference Samples

Filter

Web

- Sign in with a pre-built UI
- Get Started
- Manage Users
- Password Authentication
- Email Link Authentication
- Sign in with Google
- Facebook Login
- Sign in with Apple
- Twitter
- GitHub
- Microsoft
- Yahoo
- Phone Number
- OpenID Connect

3. Send the authentication link to the user's email, and save the user's email in case the user completes the email sign-in on the same device.

Web modular API

```
import { getAuth, sendSignInLinkToEmail } from "firebase/auth";

const auth = getAuth();
sendSignInLinkToEmail(auth, email, actionCodeSettings)
  .then(() => {
    // The link was successfully sent. Inform the user.
    // Save the email locally so you don't need to ask the user for it again
    // if they open the link on the same device.
    window.localStorage.setItem('emailForSignIn', email);
    // ...
  })
  .catch((error) => {
    const errorCode = error.code;
    const errorMessage = error.message;
    // ...
  });
```