

CS682 PROJECT

PROEJCT4: FORMS"LY

USER MANUAL

Shivendra Tripathi

Venkata Srinivasa Reddy Gade

Sonal Dattraj Doiphode

Sai Teja

Table of Contents

1. Title Page

- CS 682 Project 4
- User Manual
- Submitted by: Shivendra Tripathi, Venkata Srinivasa Reddy Gade, Sonal Dattraj Doiphode, Sai Teja

2. 1. Introduction

- What is Formly?
- Summary of platform features (brief)

3. 2. Accessing the Platform

- Hosted version (URL placeholder)
- Local setup
 - Clone instructions
 - `.env` setup (frontend + backend)
 - Install dependencies and run servers

4. 3. Authentication

- Login / Sign Up
- Google Sign-In
- Forgot Password
- Protected routes

5. 4. Home Dashboard

- Viewing all forms

- Create a new form
- Duplicate / Delete form
- Logout

6. **5. Form Builder**

- Edit title and description
- Add section(s) and questions
- Question types (MCQ, SAQ, LSQ)
- Drag and drop support for reordering
- Save and edit forms

7. **6. Group Settings**

- Parent and Child groups
- Add, edit, delete groups
- Theme customization per group (font, color)
- Toggle form acceptance

8. **7. Sharing the Form**

- Generate public links per group
- Form expiry management

9. **8. User Form View**

- What users see (MCQ / SAQ / LSQ layout)
- Submit responses
- Response submitted confirmation

10. **9. Dashboard View**

- Filter by group
- Section-wise breakdown:
 - MCQ → Bar Charts
 - SAQ → Word Cloud + Sentiment Pie
 - LSQ → Gauge Charts
- Comparison View

11. 10. AI Assistant

- Ask natural questions about survey results
- How to use the chat
- Examples of useful queries

12. 11. Exporting Data

- CSV of raw responses
- PDF of:
 - Full dashboard
 - SAQ sentiment + cloud section

13. 12. Contact

- Team member names
 - GitHub repo
 - Tech stack used
-

1. Introduction

Welcome to Formslly – A Smart Form Management Platform

Formslly is a modern, full-stack web application built to simplify how users create, distribute, and analyze custom surveys. Designed for both technical and non-technical users, Formslly empowers organizations to build structured forms, collect group-specific responses, and gain actionable insights through a powerful dashboard interface.

Originally developed by a previous team with core capabilities like form creation, response capture, and basic group comparison, the platform has now evolved significantly. The Spring 2025 version introduces a robust suite of new features that transform Formslly into an intelligent form analytics platform.

What You Can Do with Formslly

- **Create surveys with:**
 - Multiple Choice Questions (MCQ)
 - Short Answer Questions (SAQ)
 - Likert Scale Questions (LSQ)
- **Group Responses:**
 - Generate links for different audience segments (e.g., classes, departments, teams)
 - View comparative results across groups
- **Visualize Insights:**
 - MCQ: Bar charts
 - SAQ: Word clouds + sentiment pie charts
 - LSQ: Gauge charts with interpretation

- **Customize Experience:**
 - Set per-group dashboard themes (font, color)
 - Toggle whether a form is accepting responses
 - Duplicate or delete forms instantly
 - **AI-Powered Assistance:**
 - Built-in chatbot answers your questions about survey trends
 - Uses Grok's LLaMA model for accurate natural language analysis
 - **Export Reports:**
 - Download full response data (CSV)
 - Export dashboard summaries as PDF
-

Tech Stack Overview

- **Frontend:** React + Vite, Mantine UI, Tailwind CSS, Chart.js
 - **Backend:** Node.js, Express.js
 - **Database:** MongoDB (via Mongoose)
 - **Auth:** Firebase (Email + Google Sign-In)
 - **AI Integration:** Grok API (LLaMA 3.3 model)
-

2. Opening the Application

Users can access Formslly in **two ways** — either through a **live deployed link** or by setting up the project **locally** on their development machine.

A. Hosted Deployment (Recommended)

To access the hosted version of the application, simply visit the following URL in your browser:

formslly-eta.vercel.app/

This version is fully functional and ready to use with all features enabled — including form creation, group comparisons, dashboard insights, AI assistance, and export tools.

B. Running the Project Locally

Developers and contributors can also set up the full-stack application on their local machine using the following steps:

Step 1: Clone the Repository

```
git clone https://github.com/srinivas9059/682_4  
cd 682_4
```

Step 2: Setup Environment Variables

You must configure `.env` files for both the frontend and backend.

Frontend (inside `/front-end` folder)

Navigate to the frontend directory:

```
cd front-end
```

- 1.

Create a new file named `.env` and paste the following content:

```
VITE_BACKEND_URL=http://localhost:3000
VITE_CLIENT_BASE_URL=http://localhost:5173
VITE_API_KEY=AIzaSyCOEh2M5celoIsTPhd-AlN9v3w_-fkWH9U
VITE_AUTH_DOMAIN=forms-project-auth.firebaseio.com
VITE_PROJECT_ID=forms-project-auth
VITE_STORAGE_BUCKET=forms-project-auth.appspot.com
VITE_MESSAGING_SENDER_ID=525852451699
VITE_APP_ID=1:525852451699:web:130efe1b21a3115f4d9b06v
```

Backend (inside `/back-end` folder)

Navigate to the backend directory:

```
cd ../back-end
```

1.

Create a new file named `.env` and paste the following content:

```
PORT=3000
MONGODB_URI=your_mongodb_connection_string_here
CLIENT_BASE_URL=http://localhost:5173
GROQ_API_KEY=your_groq_api_key_here
```

2.

Replace `your_mongodb_connection_string_here` and `your_groq_api_key_here` with your actual credentials.

Step 3: Install Dependencies

Frontend:

```
cd ../front-end
npm install
```

Backend:

```
cd ../back-end
npm install
```

Step 4: Start the Servers

In two separate terminals:

Frontend:

```
cd front-end  
npm run dev
```

Backend:

```
cd back-end  
npm start
```

Access the App Locally

Once both servers are running, open your browser and go to:

<http://localhost:5173>

This launches the local version of Formsls where all features — including form creation, group filtering, AI analysis, and dashboard export — are available for testing or development.

3. User Registration and Login

Before using Formslly, users must create an account or log in using one of the supported authentication methods. The system uses **Firebase Authentication**, ensuring secure and scalable user identity management.

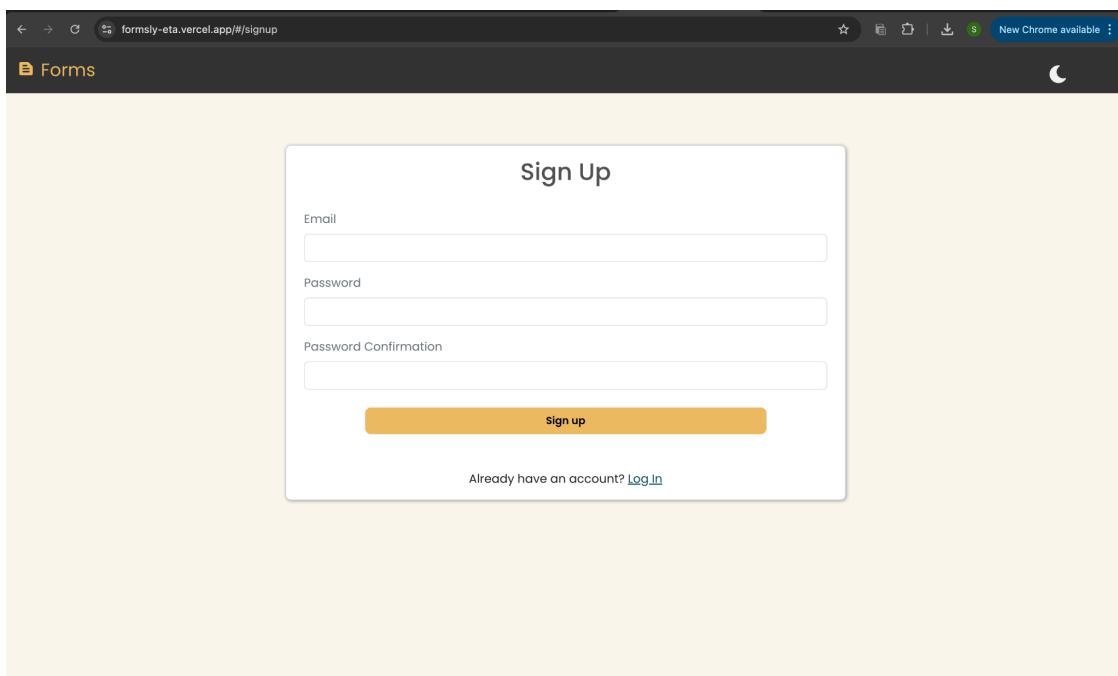
A. Sign Up (Register for an Account)

To create a new account:

1. Visit the live site at
formslly-eta.vercel.app/
2. Click “**Sign Up**” in the top navigation bar or on the login page.

3. Fill in the **Email**, **Password**, and **Confirm Password** fields.

4. Click the “**Sign Up**” button.



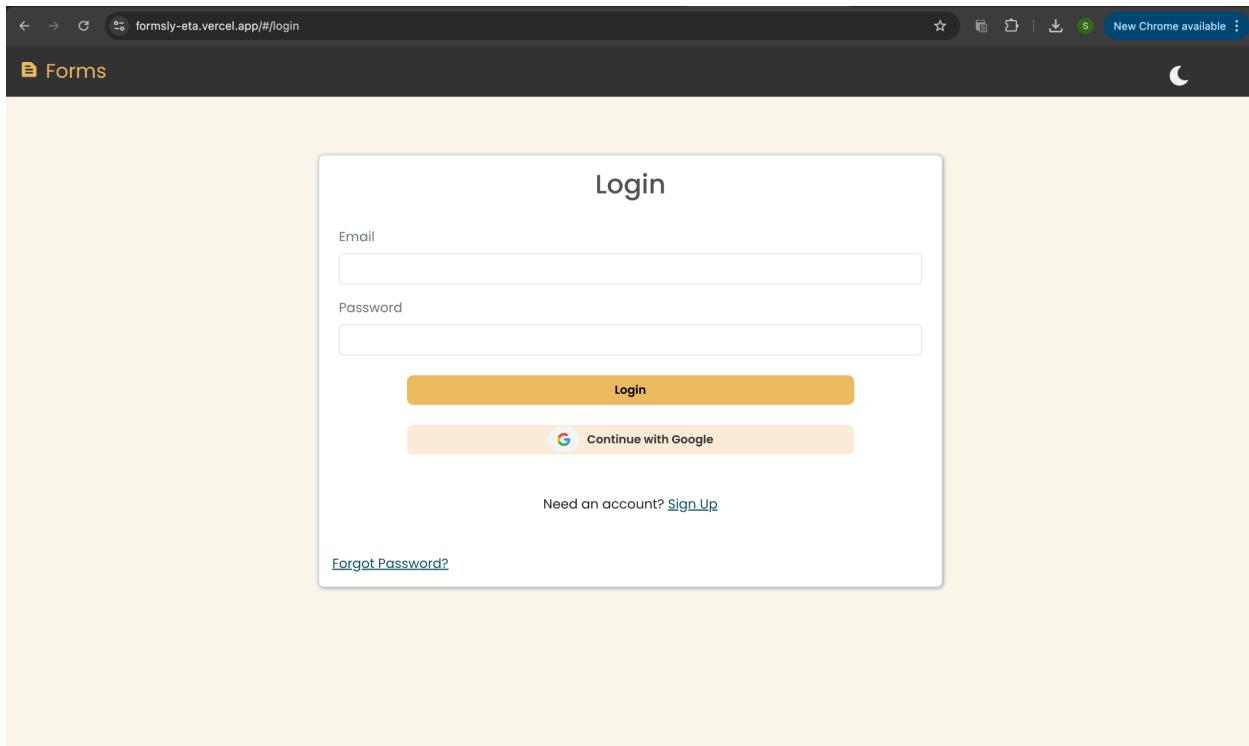
B. Login with Email & Password

To log in with credentials:

1. Click “**Login**” on the homepage or nav bar.
2. Enter your **email** and **password**.
3. Click “**Login**”.

On success, you’ll be redirected to your **Forms Dashboard**.

Invalid credentials or server issues will trigger an error message.



C. Login with Google (OAuth)

You can also log in using your **Google Account**:

1. On the **Login page**, click “**Sign in with Google**”.

2. Select or log into your Google account in the popup.
3. You'll be automatically redirected to your dashboard.

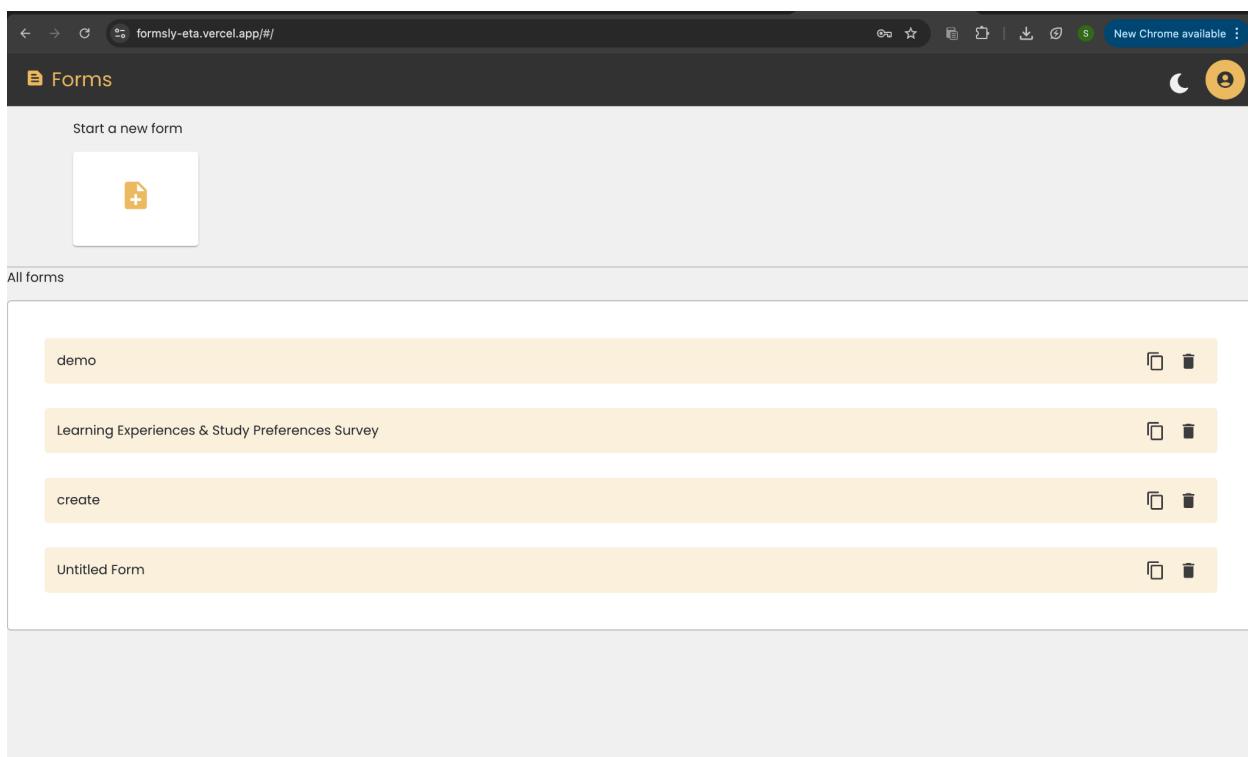
Make sure your browser allows pop-ups for Google sign-in to work properly.

4. Home Dashboard

After logging in, you are redirected to the **Home Dashboard**, where you can view and manage all forms you've created. This page acts as the control center for form lifecycle management.

A. Viewing All Forms

- The dashboard displays a **list of all forms** created by the logged-in user.
- Each item in the list shows:
 - **Form Title**
 - **Unique Form ID**
 - Icons for actions: Duplicate, Delete
- Data is fetched via: `GET /getAllFormTitlesIDs?userID=...`

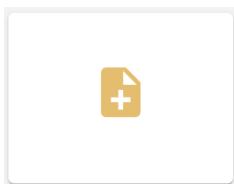


The screenshot shows a web browser window with the URL `formslly-eta.vercel.app/#/` in the address bar. The page title is "Forms". On the left, there's a button labeled "Start a new form" with a plus sign icon. Below it, a section titled "All forms" lists four entries:

Form Title	Action Buttons
demo	[Duplicate] [Delete]
Learning Experiences & Study Preferences Survey	[Duplicate] [Delete]
create	[Duplicate] [Delete]
Untitled Form	[Duplicate] [Delete]

B. Create a New Form

- Click the “Start a New Form” card (+) to initiate a new form.



- This triggers a call to GET /createForm.
- A new form with default settings is created and saved.
- The app automatically redirects you to the **form builder** for editing.

C. Duplicate or Delete a Form

Duplicate:

- Click the **duplicate icon** beside a form.



- A clone of the form (without responses) is created via `GET /duplicateForm/:id`.
- The duplicate is added to your form list.

Delete:

- Click the **trash icon** beside a form.



- The form is permanently deleted using `DELETE /deleteForm/:id`.
- Confirmation is shown via an on-screen notification.

D. Logout

- Click the **Account Icon** in the top-right corner.
- Select **Logout** to end your session.
- The app calls `logout()` from the Firebase AuthContext and redirects to `/login`.

The screenshot shows a web application interface for managing forms. At the top, there is a header bar with a back arrow, forward arrow, refresh icon, and a search bar containing the URL "formly-eta.vercel.app/#/". To the right of the search bar are several browser icons and a message "New Chrome available". A user profile dropdown menu is open, showing "Update Profile" and "Log out".

The main area is titled "Forms" and contains a button "Start a new form" with a plus sign icon. Below this is a section titled "All forms" which lists four entries:

- demo
- Learning Experiences & Study Preferences Survey
- create
- Untitled Form

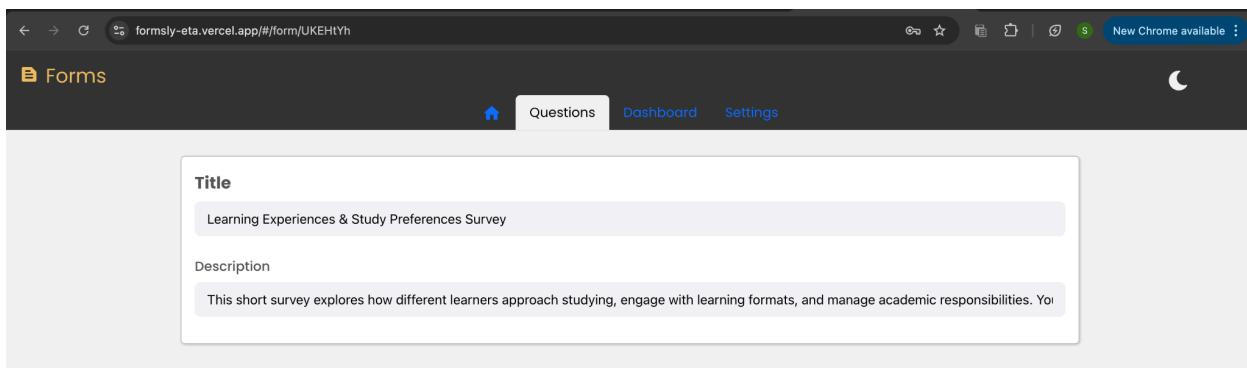
Each entry has a small icon to its right, likely for editing or deleting.

5. Form Builder

The **Form Builder** lets you create custom forms with multiple sections and questions. It supports a modular design, real-time updates, and drag-and-drop editing.

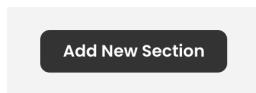
A. Edit Title and Description

- The form title and description are editable at the top of the builder.
- Live updates are synced using the `FormTitleDescription.jsx` component.
- Any change is reflected immediately in the form preview.
- The state is handled locally and synced during save.



B. Add Section(s) and Questions

- Forms are organized into **sections**. Each section can hold multiple questions.



- Sections are displayed one below the other using `Section.jsx`.
- Questions are added inside sections using a "+" button.



- Each question can be duplicated or deleted easily using action icons.

The screenshot shows a web-based form builder interface. At the top, there's a header with a back arrow, a refresh icon, and the URL 'formly-eta.vercel.app/#/form/ITFzBVb'. Below the header, there's a 'Description' section with a text input field. The main area contains a question card. The question text is 'are you listening to the class'. To the right of the question, there's a dropdown menu set to 'Multiple Choice'. Below the question text, there are two radio button options: 'yes' and 'no', each followed by a delete icon (an 'X'). At the bottom of the question card, there are edit and delete icons. At the very bottom of the screen, there are 'Add New Section' and 'Save' buttons.

C. Question Types (MCQ, SAQ, LSQ)

Each question can be of one of three types:

1. MCQ (Multiple Choice Question)

- Uses radio buttons for options.

This screenshot shows the same form builder interface as the previous one, but with a different question setup. The question text is 'Enter Question'. The dropdown menu next to the question type is also set to 'Multiple Choice'. Below the question text, there is a single radio button option labeled 'Add option', which is highlighted with a blue selection bar. To the right of the question card, there's a dropdown menu with three options: 'Multiple Choice', 'Short Answer', and 'Likert Scale'. The 'Multiple Choice' option is currently selected.

- Options are reorderable using drag-and-drop.

- Managed via `MCQuestion.jsx` and `SortableOption.jsx`.

Question

Which learning format do you prefer the most?

Multiple Choice

- In-person classes = X
- Online live sessions = X
- Recorded lectures = X
- Self-paced modules = X
- Add option

□ ✖

2. SAQ (Short Answer Question)

- A free-text input field for open-ended answers.
- Managed using `SAQuestion.jsx`.

Question

What motivates you the most to study regularly?

Short Answer

□ ✖

3. LSQ (Likert Scale Question)

- A scale from 1–5 (or customizable range) for rating.
- Each scale point can be labeled.
- Rendered using `LSQuestion.jsx`.

A screenshot of a Likert scale question component. At the top left is a "Question" header with a three-dot menu icon. Below it is a text input field containing the statement: "I feel confident about managing my study time.". To the right is a "Likert Scale" dropdown menu. The main area contains a "Upper Limit" input field set to "5". Below this is a vertical list of five response options from "Strongly Disagree" to "Strongly Agree". At the bottom is a horizontal slider with five positions labeled "Strongly Disagree", "Disagree", "Neutral", "Agree", and "Strongly Agree", each with a corresponding gray circular handle. On the far right are two small icons: a copy/paste icon and a trash bin icon.

The main question type selector is rendered inside `Question.jsx`, and determines which specific component to display.

D. Drag and Drop Support

- Reordering of questions is powered by **DndKit**:
 - Questions can be dragged using `SortableQuestion.jsx`.

A screenshot of a short answer question component. It features a "Question" header with a three-dot menu icon. Below it is a text input field containing the question: "What motivates you the most to study regularly?". To the right is a "Short Answer" dropdown menu. At the bottom are two small icons: a copy/paste icon and a trash bin icon.

- MCQ options are drag-enabled using `SortableOption.jsx`.

Question

What time of day do you usually prefer to study?

Multiple Choice

Afternoon = X

Early Morning = X

Evening Drag X

late Night = X

Add option

- Intuitive UX helps manage long forms with multiple sections easily.
-

E. Save and Edit Forms

- All changes are stored locally during editing.
- Click “Save” or navigate away to trigger an update via:



- `PUT /updateForm` for overall form.
 - `PUT /updateSection` for per-section changes.
 - Reopening an existing form loads it using:
 - `GET /getFormData/:formID`
-

6. Group Settings

The **Group Settings** section enables form creators to organize survey respondents into hierarchical groups, customize visual themes per group, and manage form availability. This helps in generating targeted links and producing segmented analytics.

A. Parent and Child Groups

- The system uses a **three-level hierarchy**:
 - **Parent Group** (`groupCode: "1"`): Top-level category (e.g., Department).
 - **Child Group** (`groupCode: "2"`): Mid-level category, nested under a parent.
 - **Leaf Group** (`groupCode: "3"`): Final shareable group used for distribution.
- Groups are stored inside the form document (`formGroups` and `formParentGroups` arrays).
- The Settings panel renders these hierarchies using:
 - `GroupCard.jsx`, `GroupListItem.jsx`, `ChildGroupCard.jsx`.

The screenshot shows the Formly app's Settings page. At the top, there are navigation links for Home, Questions, Dashboard, and Settings. The Settings tab is active. Below the navigation, there is a "Accepting Responses" toggle switch, which is turned on. The main content area is titled "Groups". On the left, there is a tree view of groups: "Student Group" has children "Undergraduates" and "Graduates"; "Working Group" has children "Working Learners". A blue "+" button is located next to the "Student Group" node. On the right, a "Parent Group" section is displayed, showing a "Group Name" input field containing "Working Group". Below this are four buttons: "Save Changes" (black), "Add Child Group" (black), "Create Leaf" (black), and "DELETE GROUP" (red).

B. Add, Edit, Delete Groups

- **Add Group:**

- You can add a new parent group by using the "+" buttons.



ADD PARENT GROUP

Add Child Group

- Triggered via modal dialogs, with support from `handleAddParentFormGroup()` and `handleAddFormGroup()`.
- After adding a new group, select the group and go hit the save changes button.

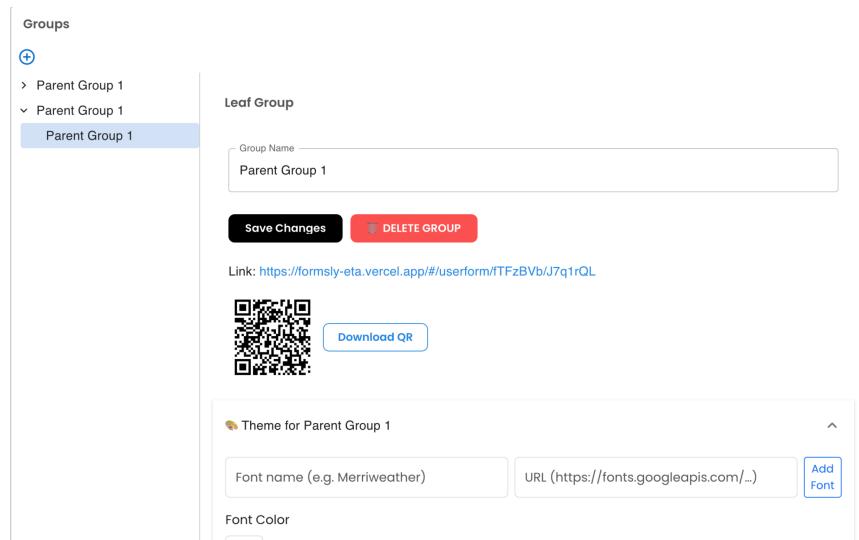
Save Changes

- **CREATE LEAF:**

- Every time you trigger create leaf it will give you new link for the form.



- After you create leaf group, you have to go trigger save changes in leaf group this time.



- **Delete Group:**

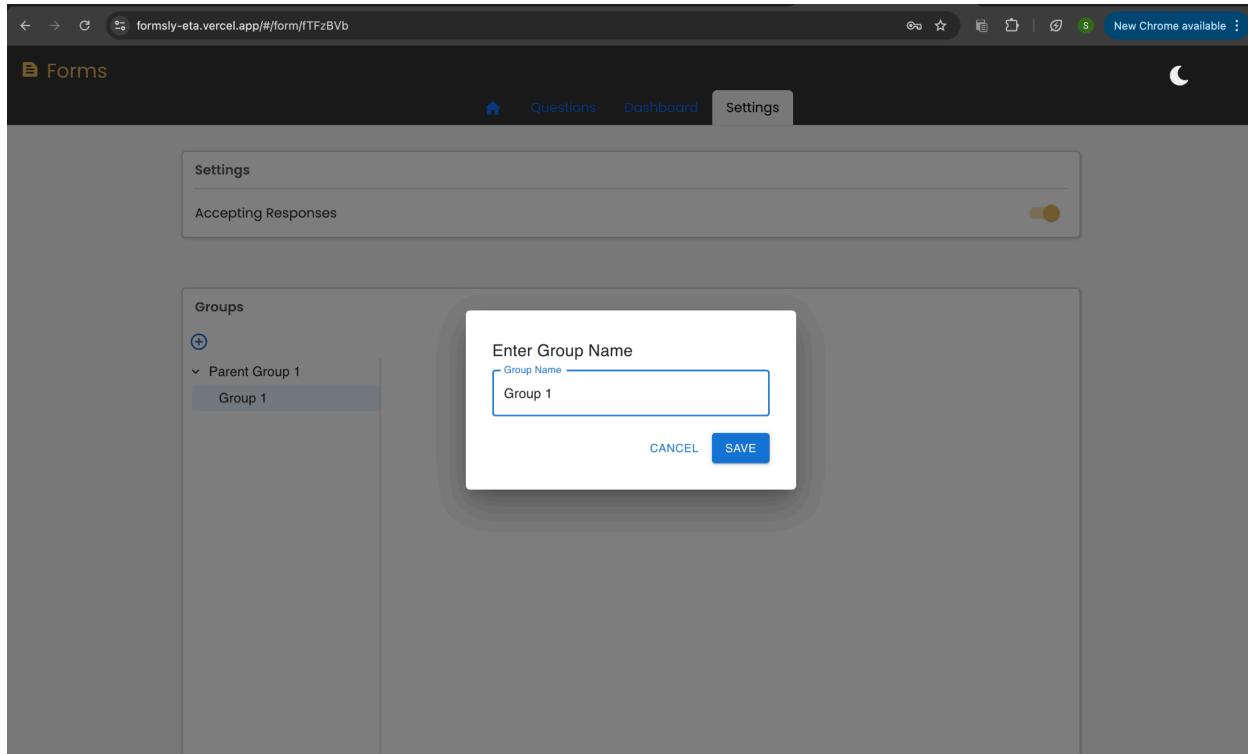
- Each group name is editable via modals from `GroupListItem.jsx`.



- You can delete group or leaf, just remember to trigger "save changes". For example if you delete a leaf, you need to go and trigger "save change" in the parent group.

- **Edit Group Name:**

- Each group name is editable via modals from `GroupListItem.jsx`.
- Changes are persisted in backend with updated form state.



C. Theme Customization per Group

- Each group can have **its own theme**:
 - **Font Family**
 - **Primary Color**
 - **Background Image URL**
- This allows you to brand forms differently for different audiences (e.g., Teams, Departments).
- Backend Support:
 - API: `PUT /updateFormGroupTheme/:formID/:groupID`

- Form schema supports per-group **theme** objects.
- Customizations reflect on the form.

The screenshot shows a user interface for managing groups. On the left, there's a sidebar with 'Groups' and a '+' button. Below it are two collapsed sections: 'Student Group' (Undergraduates, Graduates) and 'Working Group' (Working Learners). The 'Working Learners' section is expanded, showing its details. To the right, a QR code is displayed with a 'Download QR' button. Below the QR code is a panel titled 'Theme for Working Learners' with the following fields:

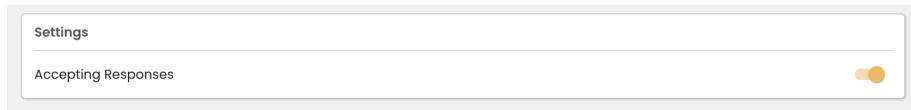
- Font name (e.g. Merriweather)
- URL (<https://fonts.googleapis.com/>...)
- Font Color (a black square)
- Font Family (set to Arial)
- Background Image URL (<https://...>)

At the bottom right of the panel are 'Reset' and 'Save Theme' buttons.

D. Toggle Form Acceptance

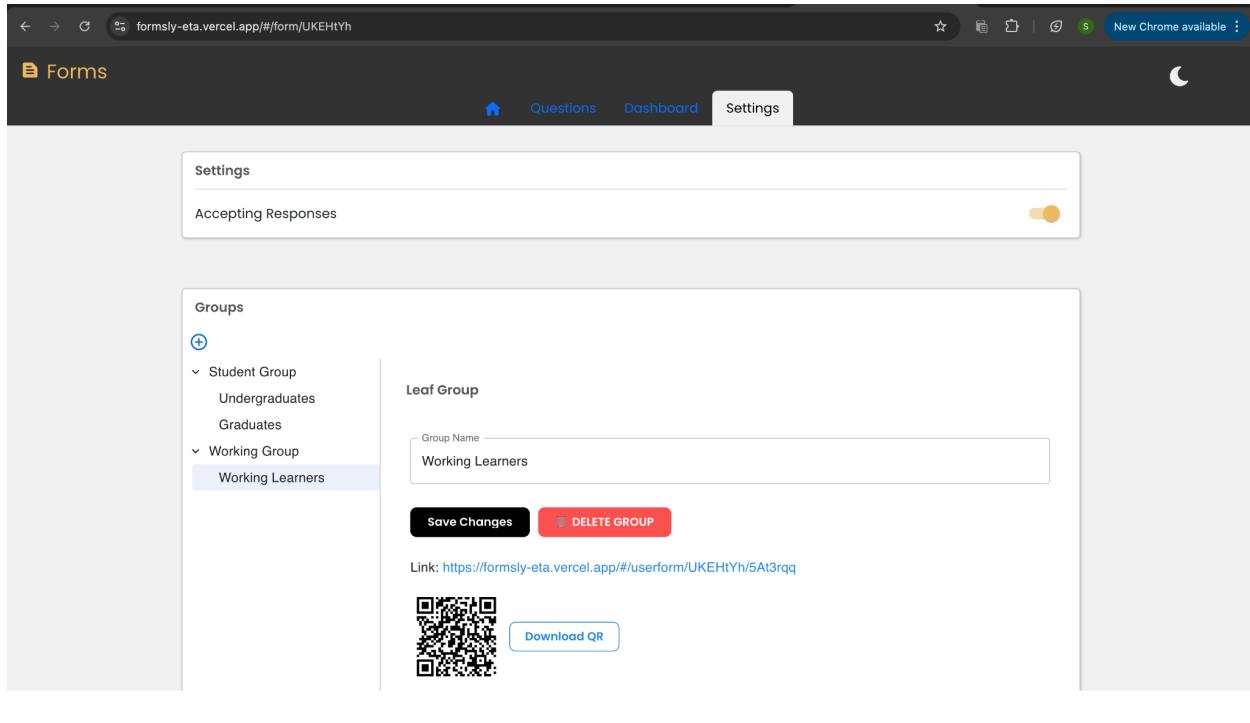
- Admins can choose to **enable or disable submissions** for a form:

- Toggle switch is available in **Settings.jsx**.



- Uses the **formIsAcceptingResponses** flag in the database.
- Backend Routes:
 - `POST /setIsAcceptingResponses` → to update the flag.

- GET /getFormIsAcceptingResponses/:id → to check current status.
- If disabled, users are redirected to the `NAResponses.jsx` screen.



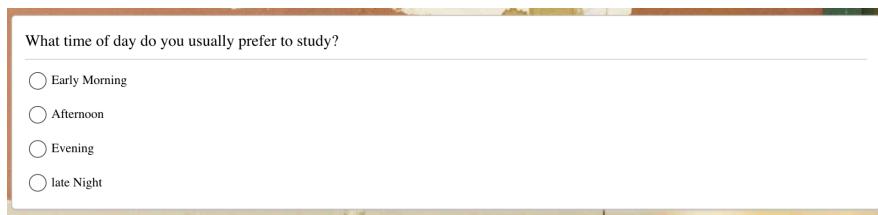
8. User Form View

The **User Form View** is the public-facing survey interface that respondents use to fill out and submit their answers. It is dynamically rendered based on the form's structure and group-specific access link.

A. What Users See (MCQ / SAQ / LSQ Layout)

When a user accesses the form through a unique group link (e.g., `/userform/:formID/:groupID`), the system:

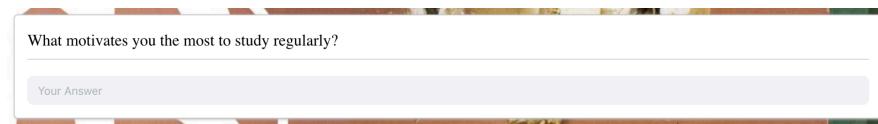
- **Fetches the form content** (title, description, questions) using:
 - `GET /getFormData/:id`
- **Renders questions dynamically** based on `questionType`:
 - **MCQ (`questionType: 1`)** → Displayed using radio buttons (`UserFormMCQ.jsx`)



What time of day do you usually prefer to study?

Early Morning
 Afternoon
 Evening
 late Night

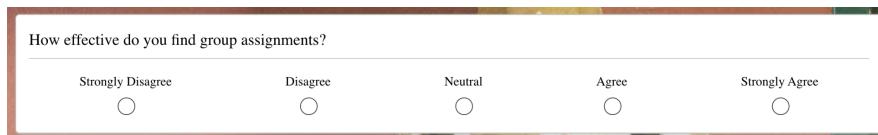
- **SAQ (`questionType: 2`)** → Displayed using a text input (`UserFormSAQ.jsx`)



What motivates you the most to study regularly?

Your Answer

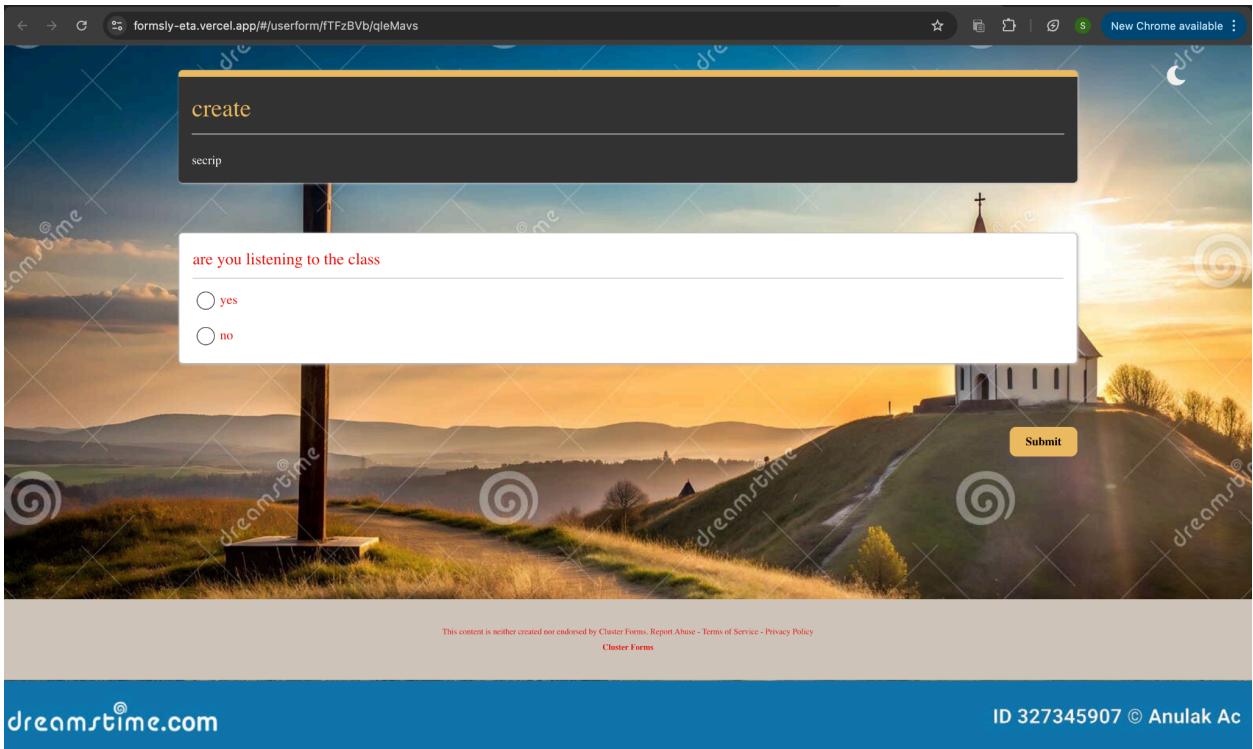
- **LSQ (`questionType: 3`)** → Displayed using a Likert scale radio group (`UserFormLSQ.jsx`)



How effective do you find group assignments?

Strongly Disagree Disagree Neutral Agree Strongly Agree

- Users also see the form title and description at the top.
- The UI adapts group-specific **theme styles** (font, colors) if configured.



B. Submit Responses

Once all questions are answered:

- The user clicks **Submit**.



- This triggers:
 - `handleSubmit()` in `UserForm.jsx`
 - Sends data to backend using:

- POST /saveUserFormResponse
- Payload includes `formID`, `userGroupID`, and an array of `{questionID, answer}` pairs.
- Form responses are saved in the `formResponses` array inside the form document.

What motivates you the most to study regularly?

Your Answer

I feel confident about managing my study time.

Strongly Disagree Disagree Neutral Agree Strongly Agree

What time of day do you usually prefer to study?

Early Morning
 Afternoon
 Evening
 late Night

Submit

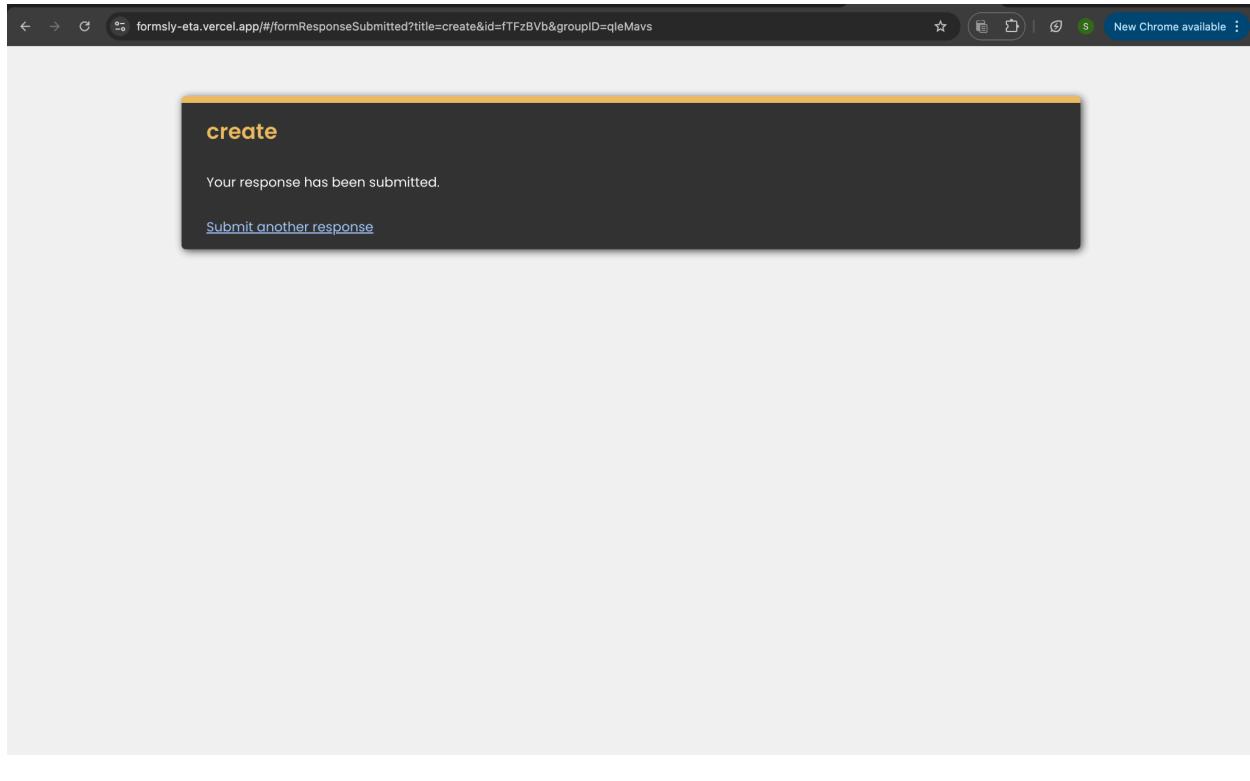
C. Response Submitted Confirmation

After successful submission:

- The user is redirected to `/responseSubmitted`.
- This screen:
 - Shows the **form title**
 - Displays a message confirming successful submission
 - Offers a **link to submit again** for the same group/form

- Built using: `ResponseSubmitted.jsx`
- Data is passed via query params (title, id, groupID)

If the form is **not accepting responses**, the user is redirected to the **NAResponses** screen instead.

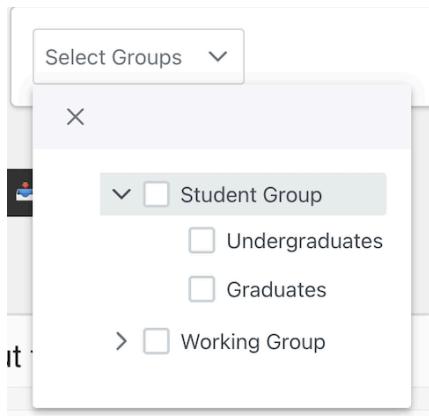


9. Dashboard View

The **Dashboard** is the analytics section where admins can view detailed summaries of form responses. It supports group-based filtering, question-type-specific visualizations, and downloadable insights.

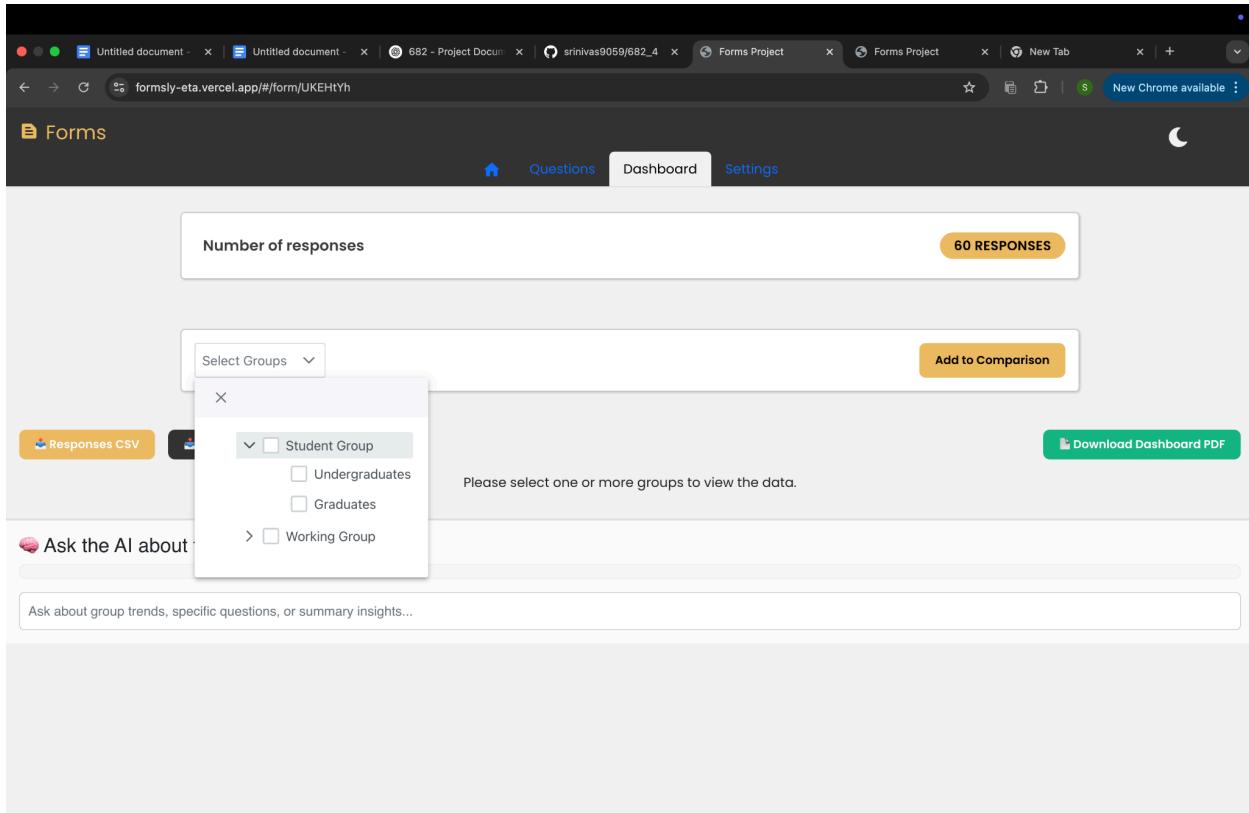
A. Filter by Group

- Admins can **filter responses** by:
 - **Parent groups** (e.g., Student, Working)
 - **Child or Leaf groups** (e.g., Undergraduates, Graduate, Working Learners)
- Group filters are implemented using a **TreeSelect dropdown**.



- Allows:
 - Single group view
 - **Comparison View** between two selected groups (by using the button below)

Add to Comparison

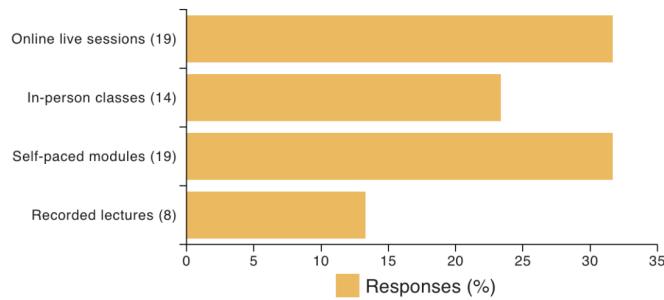


B. Section-wise Breakdown

For each section and question, the dashboard dynamically renders charts depending on the **question type**:

MCQ (Multiple Choice Questions)

- **Component:** `MCQDashboardListItem.jsx`
- **Visualization:** Bar Chart (via `@mui/x-charts`)
- **Features:**
 - Counts displayed per option
 - Comparative view shows both groups side-by-side
 - Text summary of most and least selected options



Online live sessions was the most preferred choice, receiving **31.67%** of responses. It was followed by **Self-paced modules** at **31.67%**. The least chosen option was **Recorded lectures**, making up only **13.33%** of responses.

SAQ (Short Answer Questions)

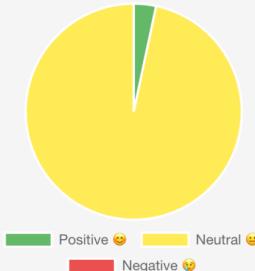
- **Component:** `SAQDashboardListItem.jsx`
- **Visualizations:**
 - **Word Cloud** (raw or keyword-based using NLP)
 - **Sentiment Pie Chart** (positive, neutral, negative)
- **Features:**
 - Toggle between raw words and AI-extracted keywords
 - Export full section as PDF (word cloud + pie)
 - Keyword extraction via `compromise` library
 - Sentiment analysis via `sentiment` npm package

What motivates you the most to study regularly?
60 responses

[Download \(PDF\)](#)

Sentiment Analysis (Parent Group):

- 😊 Positive: 3.33%
- 😐 Neutral: 96.67%
- 😢 Negative: 0.00%

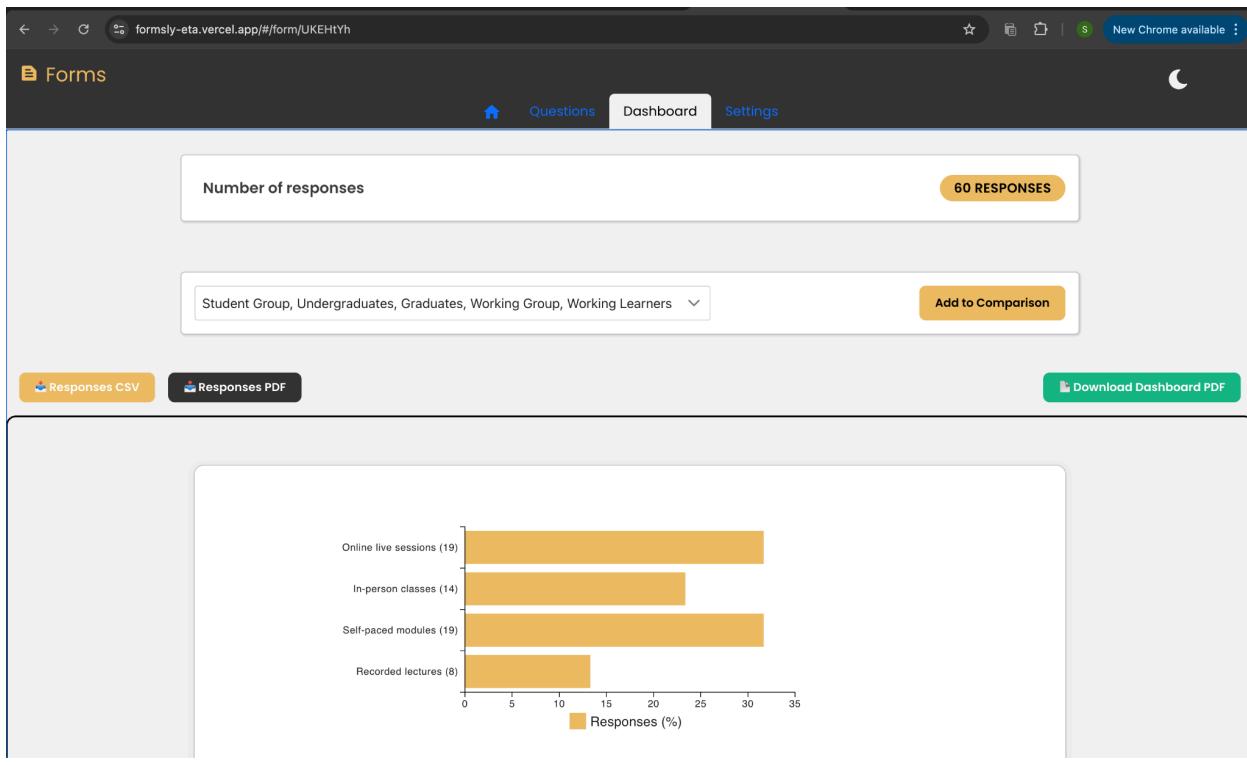
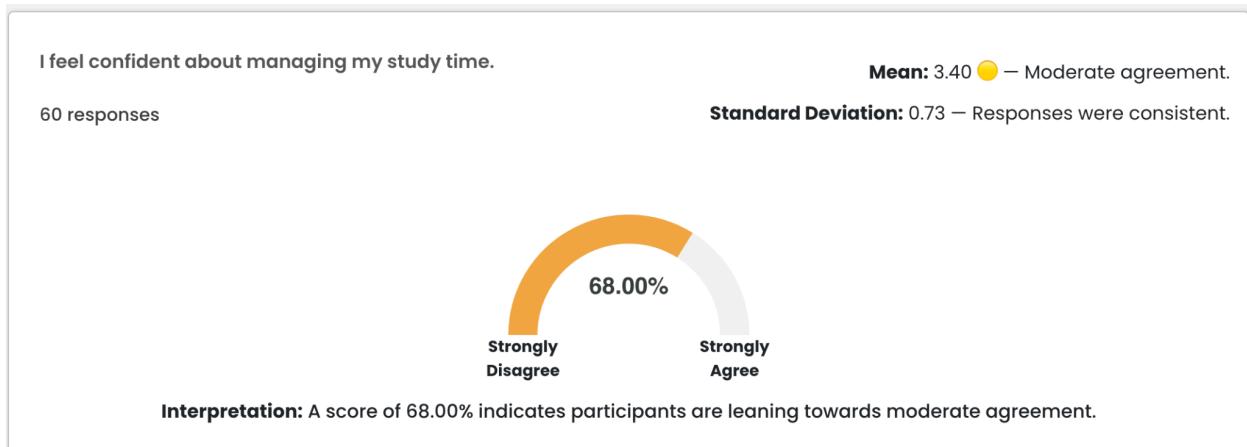


security friends job shift future-proofing upgrading internship field grades
career promotions promotion getting studying interest building skills skill internships

Show Only Keywords

LSQ (Likert Scale Questions)

- **Component:** `LSQDashboardListItem.jsx`
- **Visualization:** Half-circle Gauge (via `ReactApexChart`)
- **Features:**
 - Gauge shows average sentiment in %
 - Vertical labels for “Strongly Disagree” → “Strongly Agree”
 - Comparison view displays two gauges side-by-side
 - Textual interpretation of mean and standard deviation



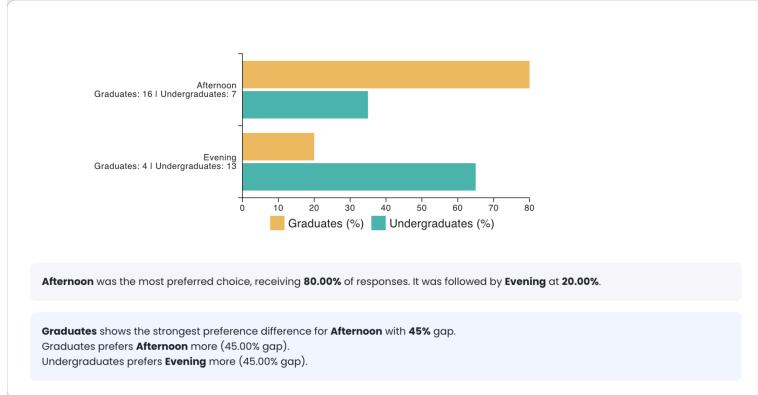
C. Comparison View

- Enabled by selecting **two groups** from the dropdown.

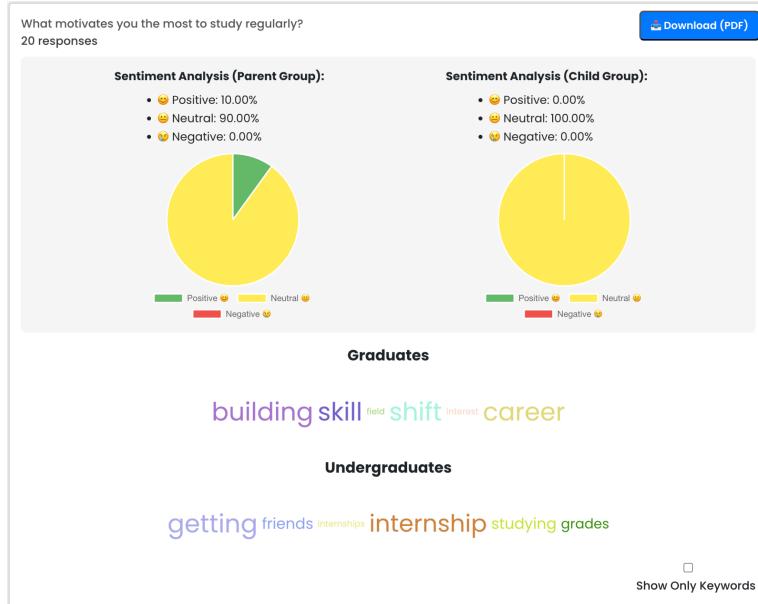
Add to Comparison

- Renders side-by-side comparisons for:

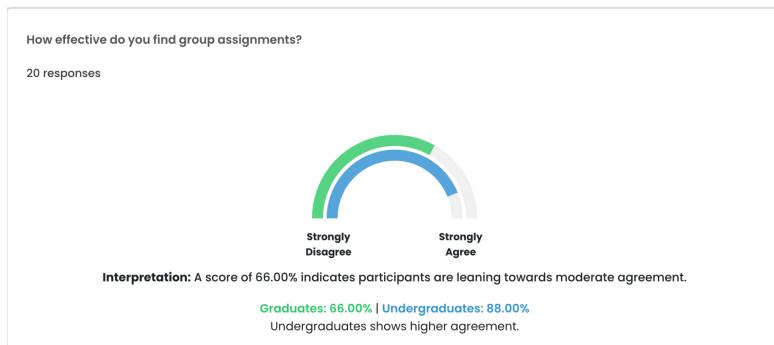
- **MCQ** → Dual-bar comparison



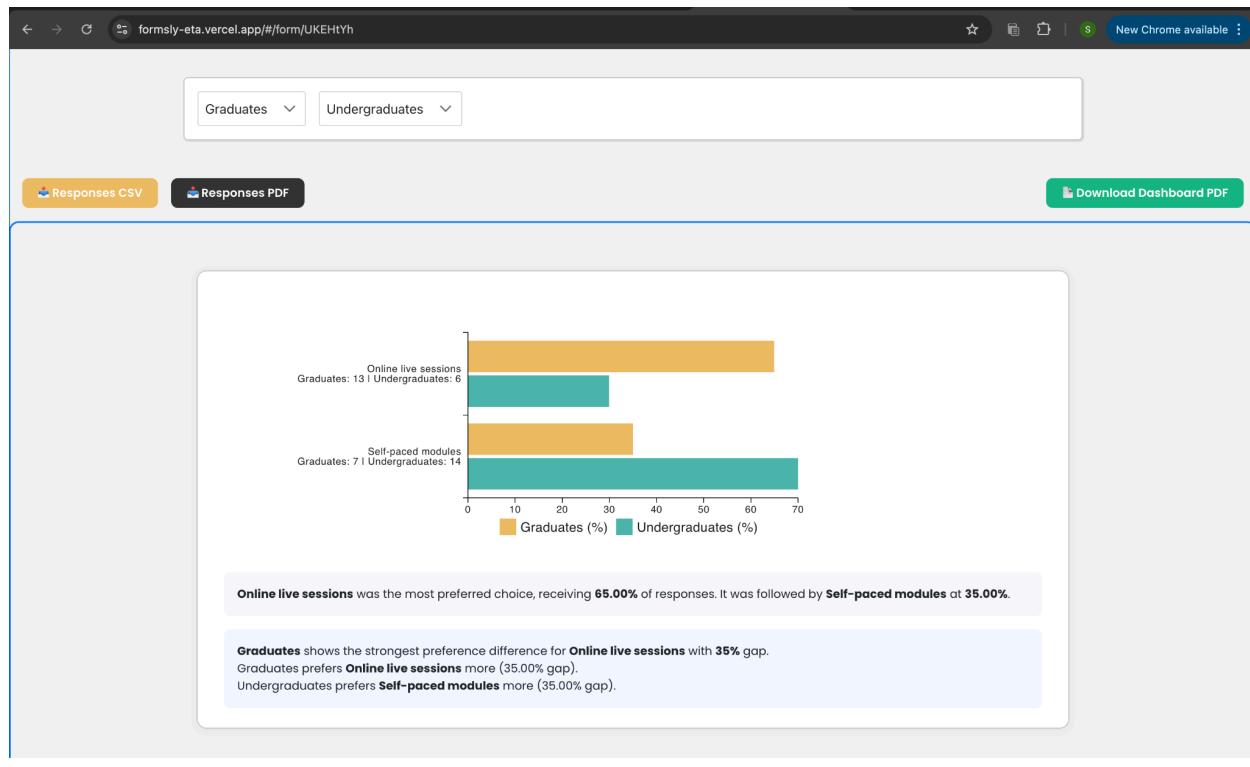
- **SAQ** → Two sentiment pies + word cloud alignment



- **LSQ** → Twin gauge charts with aligned labels



- Useful for identifying shifts in opinion or engagement across teams, departments, or cohorts.

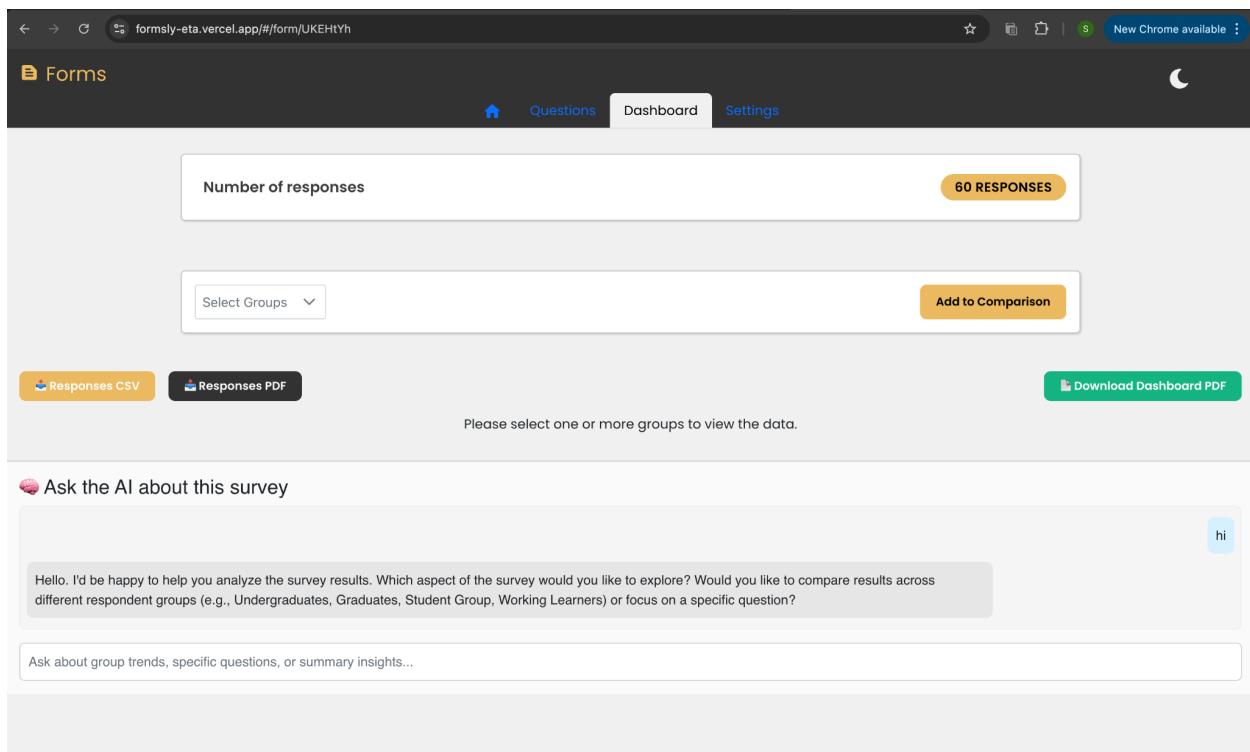


10. AI Assistant

The built-in **AI Assistant** is one of the standout features of the new version. It allows admins to ask questions about the survey results in plain English and receive **context-aware answers** powered by Groq's LLaMA 3.3 model.

A. Ask Natural Questions About Survey Results

- The assistant uses both the **form structure** (questions and groups) and the **summary data** (response statistics) to generate answers.
- It supports **comparative** and **insight-based** queries.
- No technical skills are needed — just ask like you would in a conversation.



The screenshot shows a web browser window for 'formly-eta.vercel.app/#/form/UKEHtYh'. The top navigation bar includes 'Forms' (selected), 'Questions', 'Dashboard', and 'Settings'. Below the navigation is a summary card with 'Number of responses' and a large orange button labeled '60 RESPONSES'. A dropdown menu for 'Select Groups' has 'Add to Comparison' next to it. At the bottom, there are buttons for 'Responses CSV', 'Responses PDF', and 'Download Dashboard PDF'. A message at the bottom says 'Please select one or more groups to view the data.' Below this is a section titled 'Ask the AI about this survey' with a blue icon. A message from the AI starts with 'Hello. I'd be happy to help you analyze the survey results. Which aspect of the survey would you like to explore? Would you like to compare results across different respondent groups (e.g., Undergraduates, Graduates, Student Group, Working Learners) or focus on a specific question?' A text input field below asks 'Ask about group trends, specific questions, or summary insights...'. A small blue bubble on the right says 'hi'.

B. How to Use the Chat

1. **Scroll to the bottom** of the Dashboard page — the AI Chat Window appears.
2. **Type your question** in the input bar (e.g., "Which group had more positive feedback?")
3. **Press Enter** or click the send button.
4. The assistant responds with an **intelligent summary** using the latest form data.

Tip: The assistant auto-scrolls to keep your conversation in view.

C. Examples of Useful Queries

- "What is the most common feedback from Group A?"
- "Which department had the most disagreement on the policy question?"
- "Summarize overall sentiment for Section 2."
- "What did people like about the course?"
- "Compare satisfaction between Freshmen and Seniors."

The assistant does **not guess** — it only responds based on submitted answers.

ACTUAL REPLY FOR THE PROMPT: WHAT CAN YOU TELL ME.

Based on the survey data, here are some key observations and insights:

1. ****Learning format preferences**:**

* Undergraduates prefer in-person classes (14) over online live sessions (6).
* Graduates prefer online live sessions (13) and self-paced modules (7) over in-person classes (0).

* Working Learners prefer self-paced modules (12) and recorded lectures (8) over in-person classes (0).

2. ****Motivations to study regularly**:**

* Undergraduates are motivated by getting an internship (11 mentions) and friends studying (5 mentions).

* Graduates are motivated by career shift (8 mentions) and skill building (7 mentions).

* Working Learners are motivated by future-proofing skills (10 mentions) and upgrading for promotions (6 mentions).

3. **Study time preferences**:

* Undergraduates prefer studying in the evening (13) and afternoon (7).

* Graduates prefer studying in the afternoon (16) over evening (4).

* Working Learners prefer studying late at night (15) and evening (5).

4. **Response rates and engagement**:

* The "Student Group" has no responses for any question, suggesting low engagement or a potential issue with the survey distribution.

* The "How effective do you find group assignments?" question has zero responses for all groups except Working Learners, who have 11 responses.

5. **Biases and trends**:

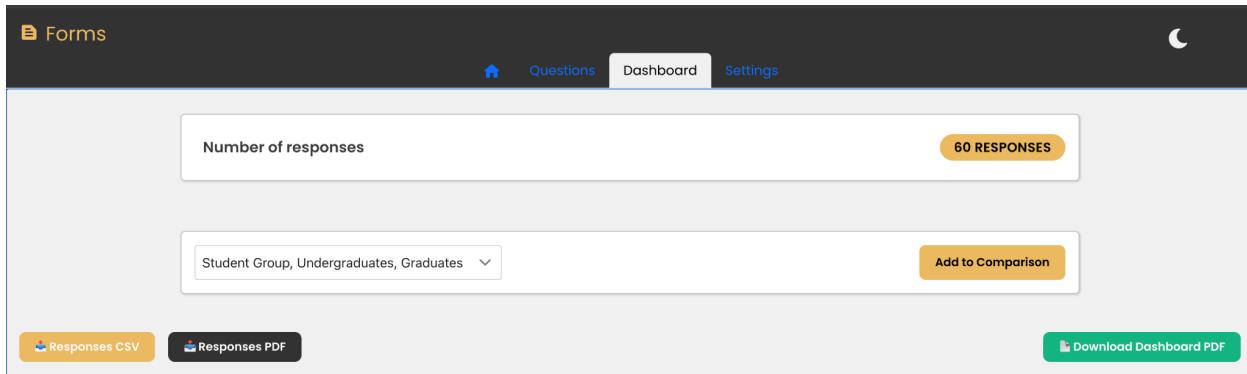
* There is a noticeable bias towards online and self-paced learning formats among Graduates and Working Learners, which may reflect their busy schedules or preferences for flexibility.

* Undergraduates seem to prioritize internships and social influences, while Graduates focus on career advancement and skill development. Working Learners are driven by the need to future-proof their skills and advance in their careers.

These observations provide a starting point for understanding the survey results and identifying patterns across different respondent groups.

11. Exporting Data

The system allows administrators to **download survey results** for offline access, reporting, or archival. Two formats are supported: **CSV (raw data)** and **PDF (visual dashboards)**.

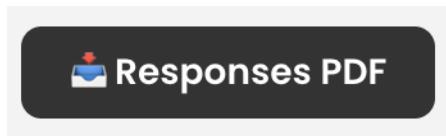


A. CSV Export – Raw User Responses

- Navigate to the **Dashboard** screen.
- Click on the “**Responses CSV**” button.



- This will trigger a download of all responses submitted across all groups.
- The same file is available in pdf format, you can download it using the button below.



- CSV format includes:
 - Group ID
 - Question ID

- User responses (MCQ, SAQ, LSQ)

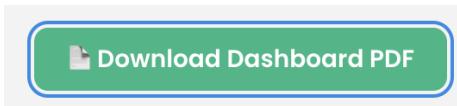
Ideal for advanced analysis in Excel, Sheets, or Python.

B. PDF Export Options

The application uses [html2canvas](#) and [jsPDF](#) to generate visual reports of the dashboards.

1. Full Dashboard PDF

- From the Dashboard, click “**Download Dashboard PDF**”.



- This includes:
 - All sections: MCQ, SAQ, LSQ
 - All selected groups (side-by-side if comparison mode is active)
 - Embedded charts and text summaries

Useful for sharing insights with stakeholders or during presentations.

2. SAQ Section Only (Sentiment + Word Cloud)

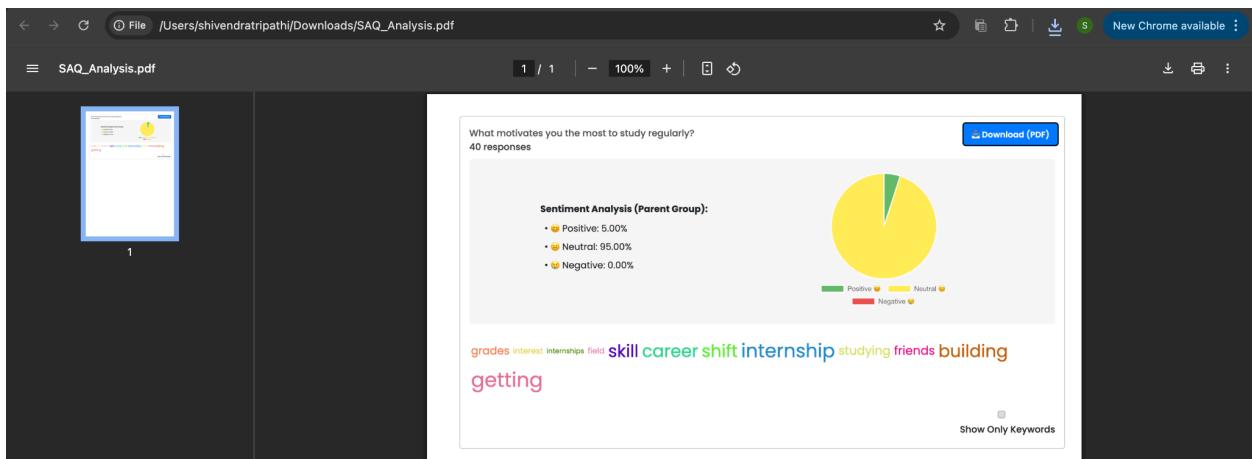
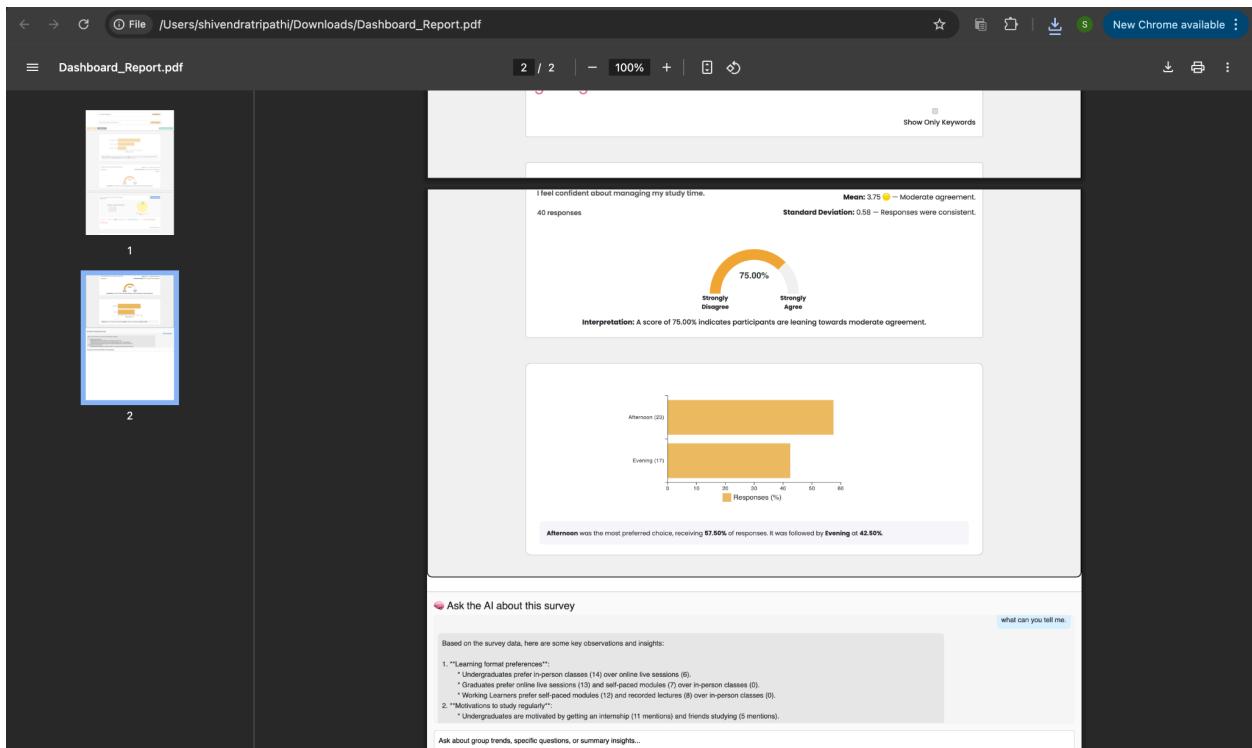
- Inside the SAQ section of the Dashboard, click “**Download (PDF)**”.



- This captures:
 - Sentiment pie chart (Positive, Neutral, Negative)
 - Keyword or raw word cloud

- Group name + question text

Best suited for qualitative data reports or open-ended feedback summaries.



13. Contact

This project was developed as part of **CS 682** during **Spring 2025**.

Team Members

- **Shivendra Tripathi**
 - **Venkata Srinivasa Reddy Gade**
 - **Sonal Dattraj Doiphode**
 - **Sai Teja**
-

GitHub Repository

You can access the full source code, documentation, and deployment scripts at:

https://github.com/srinivas9059/682_4

Tech Stack Used

Frontend:

- React (with Vite for fast development)
- Tailwind CSS + Mantine UI
- Chart.js, ReactApexChart, MUI Charts
- Firebase Authentication (Google Sign-In)
- html2canvas, jsPDF for PDF generation

Backend:

- Node.js with Express
- MongoDB with Mongoose
- dotenv for environment configs
- node-fetch for Groq API (AI)

AI Integration:

- **Grok API** – LLaMA 3.3 model for survey insight generation
-

AI Usage Acknowledgment

Some parts of this documentation were developed using ChatGPT to assist in writing structured, clear, and well-formatted technical content. All AI-generated responses were carefully reviewed and edited by the team to ensure accuracy and relevance.