Setup guide

Everything below is done in a web browser. No terminal, no Node, no npm, no credit card. Budget about 15 minutes for the first run.

Part 1 — Create the Firebase project

1. Make the project

Go to https://console.firebase.google.com and sign in with any Google account.

- **Add project** → name it (e.g. `sargam-fest`) → Continue
- Google Analytics: **turn it off**. You do not need it and it adds a step.
- Create project → wait about 30 seconds → Continue

You stay on the free **Spark** plan throughout. If Firebase ever asks you to upgrade to Blaze, you have wandered into Cloud Functions or Storage — neither is used here. Back out.

2. Turn on Firestore

- Left sidebar → **Databases and Storage → Firestore** → Create database
- Edition: **Standard**
- Location: pick the one closest to you (`asia-south1` for India, `europe-west1` for Europe). **This cannot be changed later.**
- Start in **production mode**. Rules get replaced in step 5 anyway.
- Enable

3. Turn on Authentication

- Left sidebar → **Security → Authentication** → Get started
- **Email/Password** → toggle **Enable** → Save

Leave "Email link (passwordless)" off. Nobody in this app has a real email address — accounts use internal addresses like `judge-3@festlogin.local` that never receive mail.

4. Copy your config into `config.js`

- Click **Project Overview** in the left sidebar, then the **+ Add app** button just under your project name
- Choose the web icon `</>`
- App nickname: anything. Leave "Also set up Firebase Hosting" unticked — you are deploying elsewhere in Part 2 → **Register app**
- You now see a code block containing `const firebaseConfig = { … }`
- Copy the six values into `config.js` in this folder, replacing every `PASTE_…` placeholder.

Already registered an app and need the values again? **Settings** in the left sidebar → **Project settings** → scroll to **Your apps → SDK setup and configuration → Config**.

Save the file. This is the only file you edit.

> These values are public. Anyone can read them by viewing your page source, and that is fine — they identify your project, they do not grant access to it. All access control lives in the rules you paste in the next step.

5. Paste the security rules

This step is what makes the app safe. Do not skip it.

- **Firestore → Rules** tab
- Delete everything in the editor
- Open `firestore.rules` from this folder, copy the whole file, paste it in
- **Publish**

6. Add the database indexes

The short version: you can skip this entire step. When the app runs a query that needs an index it does not have, Firestore refuses it and the error carries a link that creates exactly the right index in one click. Following that link when it appears is quicker and less error-prone than typing these tables in by hand, and it never creates one you do not need.

Create them up front only if you would rather not hit an error mid-fest.

- **Firestore → Indexes** tab → Create index
- If you are asked to choose an index type, pick **Structured**. (Vector indexes are for AI embedding similarity search and are not used here.)
- Query scope is **Collection** for all of them.

Collection ID	Fields, all Ascending	Needed for
`registrations`	`eventId`, `houseId`	Registration — always
`scores`	`eventId`, `judgeUid`	Judging — always
`substitutions`	`houseId`, `registrationId`, `status`	Only if you enable substitutions
`eventMaterials`	`houseId`, `registrationId`	Only if you enable event material

Each takes a minute or two to build.

Part 2 — Put the app online

Pick one. All three are free and need no card.

Option A — Netlify Drop (fastest, about 60 seconds)

1. Go to https://app.netlify.com/drop
2. Drag this entire folder onto the page
3. You get a live URL immediately

To update later, drag the folder again. Create a free account if you want to keep the same URL and set a custom name.

Option B — GitHub Pages

1. Create a free GitHub account and a new **public** repository
2. **Add file → Upload files**
3. **Important:** open the project folder on your computer, select **everything inside it** — `index.html`, `config.js`, the `css` and `js` folders, and so on — and drag *those* in. Do **not** drag the outer folder itself.
4. Commit
5. **Settings → Pages** → Source: `Deploy from a branch` → Branch: `main`, folder `/ (root)` → Save
6. Wait a minute; your URL appears at the top of that page

Getting a 404? Open your repository page. You should see `index.html` listed directly in the file list. If instead you see a single folder you have to click into first, that is the problem — the files went in one level too deep. Delete them and re-upload following step 3 exactly.

Option C — Cloudflare Pages

1. https://pages.cloudflare.com → Create a project → Direct Upload
2. Drag the folder → Deploy

Testing on your own machine first

Opening `index.html` by double-clicking will **not** work — browsers block ES modules loaded from `file://`. Use any of the three options above, or if you happen to have Python installed, run `python3 -m http.server` in this folder and open http://localhost:8000.

Part 3 — First run

1. Open your live URL. It redirects to **Set up your fest**.
2. Enter the fest name, pick the **timezone the fest runs in** (your device's is suggested, but change it if the fest is elsewhere — daylight saving is handled automatically), and choose an Admin password.
3. Set a separate **delete-everything password**. It guards the Danger Zone. Be clear-eyed about what it does: it stops an unattended machine and a misclick. It cannot stop someone already signed in as Admin who is determined to bypass a check running in their own browser.
4. Passwords are 3 to 8 characters. **Write the admin one down** — there is no "forgot password" email.

A note on short passwords: Firebase itself demands at least 6 characters, so the app appends a fixed suffix behind the scenes. A 3-character password is genuinely easy to guess, which for a House Manager account means someone could register or withdraw entries as that house. Use something longer for the Admin account at least.

You are now signed in as Admin.

Recommended configuration order

Work through these in order — later screens depend on earlier ones.

#	Screen	What to do
1	Settings → Fest details	Grades, max judge score, registration window. Also where you rename "House" to Team or Zone if you want to, and where the optional extras live
2	Settings → Categories	Sub-Junior, Junior, Senior, and so on. If you want categories assigned automatically, set each one's class range or date-of-birth range here
3	Settings → Points & grades	Rank ladder for each of the four event classes, plus the shared grade points
4	Settings → Participant limits	Caps on how many events one participant may enter. Optionally different per category — a participant is always measured against **their own** category
5	Settings → Entry constraints	Optional. "At most N of these events" over a set you draw by hand — no cap can express that
6	Settings → Leaderboard	Which point pools count towards the Student Talent board, plus any extra named boards
7	Accounts	Create Houses, Judges, Co-Admins and Stage Managers. Each gets a password you hand over directly. Houses can own a chest number range (Blue 100–199) so a chest number reads as a house
8	Events	Add competition items, or import a CSV
9	Participants	Add students, or import a CSV
10	Schedule	Venues and timings, and where you assign a Stage Manager to a venue on a given day and time — a single account can hold several such assignments, and the screen refuses two that overlap. Leave "visible to public" off until you are ready
11	—	Give House Managers their passwords; registration is open

Leave the optional features off for a first fest. Every one of them is listed in the README and each is a single switch you can flip later — turning one on mid-fest changes nothing that already happened.

Part 4 — Running the fest

Before the fest

- House Managers log in and register their participants for events.
- Maximum caps block registration outright, with a message naming the cap. Minimums never block — run **Downloads → Compliance report** after the

deadline to find who is short.
- Admin or Co-Admin adds entries directly too, for latecomers.

On the day

1. **Registrations** → pick the event → **Assign code letters**. This shuffles the entries and gives each a letter. Do this once registration for that event is settled — assigning letters locks House Managers out of withdrawing.

A **Stage Manager** account can do this too, from their own panel, along with ticking each entry in as it goes on so nobody is called twice. That tick is a running-order aid only — it is *not* an absence, and it is stored separately so it can never be mistaken for one. Only a judge or Admin marks somebody Absent, which is the thing that moves points.
2. **Registrations** → **Assign judges** for the event.
3. Judges log in on their phones and score. Scores save as they type. A judge can check **My schedule** in their own panel to see when and where each of their assigned events runs. For an event set to *direct* mode they pick a placement from a dropdown instead of entering a mark. If an entry does not turn up, either the judge or Admin marks it **Absent** — never leave the score blank and never enter zero for a no-show, because zero is a real score that earns a grade.
4. **Judging** → Admin can fill in for a missing judge or correct any submitted score. The original value is kept for audit.
5. **Judging** → **Finalize event**. This averages, ranks and grades. It refuses if any entry has neither a score nor an absence mark.
6. **Results** → tick the events to release → **Preview impact** to see what it does to house standings → **Publish**.

Only an Admin can publish. A Co-Admin can do everything up to that point.

What the public sees

Page	Shows
`/#/results`	House rankings, Student Talent board, per-event tables
`/#/lookup`	Search by chest number or name
`/#/schedule`	The programme, once you switch it visible
`/#/slideshow`	Auto-rotating display for a projector

Nothing appears until it is published.

Part 5 — Everyday questions

Someone's score was wrong.
Fix it in Judging, then Finalize again. If the event was already published, the public page updates automatically.

A judge did not show up.
Admin types their scores in the Judging screen. It counts exactly as if the judge had entered them.

An event has no judges at all.
Fine. The Judging screen gives Admin a single column to fill in.

We need to undo a publish.
Results → Unpublish. It disappears from the public pages immediately.

A house is disputing a published result.
Only if you enabled Settings → Appeals. A House Manager then gets an Appeals tab and can dispute any of their own published results within the window you set, attaching a screenshot as proof the appeal fee was paid — there is no payment gateway on the free tier, so a screenshot stands in for a receipt.

You decide each one **Upheld** (the result stands) or **Overturned** (it was wrong), with a written reason the house can see. Deciding an appeal does not itself change any score: correct an overturned one afterwards with a Score Override in Judging, or an Adjustment. That separation is deliberate — every

route that moves points carries a reason, and an appeal decision that quietly moved points on its own would not.

A house may have a limited number of appeals open at once (you set the number). An appeal that is **Overturned** stops counting against that limit, so a house that turns out to be right is never blocked from raising the next one.

The window opens by itself the moment a result is published and closes by itself — there is no switch to remember. Note that events published *before* you upgraded carry no publish timestamp, so their window reads as closed; publish such an event again to open one.

How do I add points manually for a house?
Accounts → Houses → Edit → Adjustment points. Negatives deduct. For anything finer, Admin → Adjustments takes a reason and applies to a house or a single participant.

Someone forgot their password.
Accounts → find them → **Password** → set a new one and tell them directly. Their name, record and data all stay the same.

Behind the scenes this issues a fresh login rather than editing the old one, because a browser is not allowed to change another user's password — that needs a server. The side effect is a leftover entry in Firebase console → Authentication, which is inert and can be ignored or tidied up later.

How do I make certificates, posters or ID cards?
Certificates → pick a template → arrange it on the canvas → Save → Generate. The editor has layers, drag-and-drop, a properties panel and placeholder tokens like `{name}` and `{results}` that fill in per participant. Photos appear wherever a participant has one, and fall back to a neutral silhouette where they do not.

The Participant ID Card design works exactly the same way — it is a design like any other, not a separate mode. **Generate always asks two separate questions: which design (decided by which one you opened), and who to produce it for.** Looking for "ID card" inside the who-for list finds nothing, because that list only ever names a population:

Who to produce for	Needs a publish first?
Every registered participant	No — print these any time after registration
Participants in published events	Yes
Winners only — people who placed	Yes
One event — every rank on one page	Yes, for that event

"Every registered participant" against the ID Card design is what makes a full set of ID cards.

Rank selection is multi-select and comes from your rank ladder, so a fest awarding a fourth place is not cut off at 3rd.

Why does a banner say "Public pages are updating" after I save?
Saves now finish instantly instead of making you wait — but if the save also affects a public page (renaming a house, editing the schedule), that page updates a moment later in the background rather than holding up your click. The banner just tells you that catch-up is happening; it clears itself in a second or two. If it ever says the update **could not be** completed, press the **Publish now** button on the banner — if that also fails, it's usually the security rules (Part 1, step 5) not being current.

We need each entry to submit a song title before the event.
Events → edit the event → **Event material**, and name what you are asking for ("Song title", "Prop list"). House Managers then get a box against that entry in **Our entries**; you approve or reject each in Admin → **Event material**, oldest first. Once approved it shows to the judge beside the code letter — never with the house's name attached, even on a blind event, so asking for it costs nothing in fairness.

Can staff and house managers message each other in the app?
Only if you enabled Settings → Fest details → Messaging. An Admin or Co-Admin

starts a conversation — personal or group — with anyone across any role, and everyone in it can reply. Nobody sees a conversation they were not added to.

Be clear-eyed about what it is not: there are no push notifications, because those need a server. A message appears live while the recipient has the Messages tab open, and not otherwise. It is useful for a control room, not for reaching somebody who is not looking.

Where do I publish contact numbers for organisers or houses?
Settings → Fest details → "Organiser contacts" for a free-form public list — Admin, Co-Admin, anyone else the fest wants reachable — added by name, role and number. House numbers are separate: Accounts → edit a house → tick "Show publicly" against whichever number should appear (every number is private by default, since most belong to students). Either way, nothing appears anywhere until Settings → Fest details → "Contact page visible to the public" is switched on, which adds a Contact tab to the public site.

Can Admin or Co-Admin register participants for a House Manager?
Only if you enable it — Settings → Fest details → "Admin/Co-Admin may register participants on a house's behalf". What happens next depends on *when* you switch it on:

- **Before registration has started** (no registration exists yet, and the registration window above has not opened) — it just works. Staff pick a house and register directly, exactly like that House Manager could.
- **After registration has already started** — each House Manager is asked once, on their own panel, whether staff may register on their behalf at all. That is a single yes/no about the *permission*, not an approval per entry: once a House Manager agrees, every future entry staff make for that house registers immediately, with no further approvals. A house that has not answered yet, or has declined, blocks staff from registering **for that house only** — it never holds up any other house.

A House Manager can change their answer later from the same place they gave it.

Where do I see or hand out the participant ID cards?
Participants → **Chest number cards**. This prints a small card per participant with their chest number, photo, house and the events they are entered in — separate from the Certificates screen's Participant ID Card *design*, which is one of several full-page layouts you can also generate for the same purpose. Use whichever fits how you plan to print and cut them.

Two houses both entered a group event — how do I tell the entries apart?
By team name. A group entry is shown as "Red", or "Red A" and "Red B" where the event lets a house field more than one. The name is fixed when the entry is created, so a substitution part-way through does not make it look like a different entry on a sheet you already printed.

Why can I not edit an event?
Published events are locked. Unpublish it on the Results screen first — changing an event's class or category while its results are public would silently invalidate the standings on display.

A judge says their scores vanish while typing.
That was a bug and is fixed. Scores now save only when Save (or Enter) is pressed, and the table no longer rebuilds itself mid-entry.

A participant is in a General event — which limits apply?
Their own category's, always. But a General event only counts towards the **overall** limit, never towards a category's class limits. Type and Tier limits are the exception: those count every matching programme a participant enters, General included.

I changed a limit setting and it refuses to save.
Some settings change *how* entries are counted rather than just the numbers — turning per-category limits on or off, changing "split by stage", switching Type or Tier limits on. Existing counts were filed under the old scheme, so saving would leave them unreadable and a participant could quietly exceed a cap. Run **Participants → Recount limits** first, then save. Recount is safe at any time and can be run twice.

Two of my events clash — someone is needed in both.
The Schedule screen flags this automatically. It checks every venue and every

day, and warns when a participant or a judge is scheduled into two events that overlap. Press **Show all clashes** for the full list with times and venues. It is a warning, not a block: overlaps are often deliberate, and only you know whether that person is actually required at both.

Someone has dropped out after registration closed.
The House Manager opens **Our entries → Substitute** and requests a swap; an Admin or Co-Admin approves it under **Substitutions**. This works until code letters are assigned — after that the running order is fixed, so a name change is a result correction in Judging instead. If the replacement would breach one of their caps the swap is refused outright.

Where do I see substitution requests?
A count appears on the Substitutions item in the menu, and the dashboard warns while any are waiting. Requests are time-critical — the window closes once code letters are assigned for that event.

How do I print one certificate for someone who missed the batch?
Certificates → the design → **Print one**, then search their chest number. It prints their real published results.

Six hundred certificates crashed my browser.
Set a smaller "Pages per print window" in the Generate dialog. Large runs are split into separate print windows; save each before starting the next.

Can people use dark mode?
Yes — the toggle sits in the top bar on public pages and at the bottom of the icon rail once signed in. It remembers per browser. Printing is always light, whatever the screen shows.

How do I wipe everything and start again?
Settings → Danger zone. Type the fest name, enter your password, confirm. This deletes all data and returns you to the first-run setup screen.

Can I run two fests in one project?
No. Create a second Firebase project and deploy a second copy of the folder.

How do I reset everything for next year?
Firebase console → Firestore → delete each collection. Then reload the app; it will offer first-run setup again.

Part 6 — Staying inside the free tier

The free Spark plan allows 50,000 document reads, 20,000 writes and 20,000 deletes per day, and 1 GiB stored. A 600-participant fest uses a small fraction of that:

Activity	Rough daily cost
Registration day, 600 participants × 4 events	~6,000 writes
Fest day: 100 events, 3 judges	~11,000 writes
500 spectators checking results all day	~3,000 reads

Public pages are cheap because they read one pre-built document per event instead of every result row. That design is what keeps the numbers this low.

Watch out for: leaving the slideshow open on many screens for days (it refreshes every two minutes), and repeatedly deleting large collections (20,000 deletes per day).

Messaging is the one feature that reads continuously. It is the only part of the app that holds a live listener open rather than reading once: every new message bills a read to everyone with that conversation on screen. For a handful of organisers in a control room that is nothing. Leaving Messages open on twenty machines all day is a different matter — if you enable it, enable it for the people who need it and close the tab otherwise.

Check usage any time under Firestore → Usage.

Troubleshooting

"Could not start" on a white page.
`config.js` still has placeholder values, or Firestore is not enabled.

"Missing or insufficient permissions." / "You do not have permission."
Nine times out of ten the rules were not pasted, or were pasted from an older copy of `firestore.rules` than the code you are running. Redo step 5 with the file from *this* folder.

This is the single most common failure when upgrading rather than installing fresh: the app gains a feature, that feature needs a new collection, and the rules still live in the console describe the previous version. The symptom is always the same regardless of which feature is missing its rule, so check this before assuming the feature itself is broken.

A feature I read about is nowhere in the app.
Most of them ship switched off — see the table in the README. Nothing appears in any menu until an Admin enables it in Settings, which is deliberate: an existing fest upgrades with nothing visibly changed.

A House Manager sees no Appeals tab, or "the window has closed" on a result that was just published.
Appeals must be enabled first (Settings → Appeals). If it is on and the window still reads as closed, the result was published *before* you upgraded — it carries no publish timestamp, so there is nothing to measure a window from. Publish that event again.

"The database needs an index for this query."
Do step 6, or click the link in the browser console — it creates the exact index needed.

A banner keeps saying the public pages are behind.
This survives even a reload on purpose, so an interrupted update is never silently lost — see Part 5. Press **Publish now** on the banner. If it keeps failing, redo Part 1 step 5 (the rules) and try again.

Logging in seems to go to the wrong place.
Fixed in v8.0 — login now waits for your role to load before sending you anywhere, so the first attempt lands correctly instead of needing a second try.

The login dropdown is empty.
No accounts of that type exist yet. Admin creates them under Accounts.

Creating a judge signed me out.
It should not — the app uses a separate connection for account creation. If it happens, sign back in; the judge was still created.

A participant photo shows as a broken image.
The Drive file is not shared. In Drive: Share → General access → "Anyone with the link" → Viewer.