IMTR School Management System (Kenya)

You are a senior full-stack architect. Build a **School Management System (SMS)** for the **Institute for Meteorological Training and Research (IMTR), Kenya,** from scratch to production. Generate code, configs, and docs. Prefer clarity, modularity, and security. Use **Node.js + Express + MySQL** on the backend; **Next.js + Tailwind CSS** on the frontend (SPA feel using App Router or Pages—pick one and be consistent). Enforce **JWT in httpOnly cookies**, **bcrypt**, **validation middleware**, **RBAC**, and **audit logging**.

Global Context & Non-Functional Requirements

- **Institution**: IMTR (Kenya). Stakeholders: Students, Lecturers/Trainers, Admin, Finance/Accounts, Librarians, IT Support, External (KMD/WMO, Donors, Parents/Guardians).
- **Locale**: Africa/Nairobi (UTC+3), currency KES, date format `YYYY-MM-DD`, phone numbers align with KE standards.
- **Compliance**: Kenya Data Protection Act (2019)—data minimization, consent for PII, role-based access, audit trails, encryption in transit, proper retention and deletion.
- **Availability & Performance**: Aim for p95 API < 300ms on core endpoints at 1k RPS (scalable via horizontal scaling).
- **Security**: OWASP Top 10, CSRF defense, strict cookie flags, rate limiting, input validation, safe file uploads, secrets in env.
- **Observability**: Structured JSON logs, request IDs, health checks, metrics endpoints, error tracking.
- **DevEx**: Dockerized, reproducible dev env, seed data, make scripts, CI/CD pipeline.

1) Core Features / Modules (DDD)

Implement as **domain modules** (each with controller, service, repo/ORM, routes, validation, tests):

1. **Auth & Users**: Login/logout, refresh, password reset, email verification (optional), MFA (optional). Roles: `ADMIN`, `LECTURER`, `STUDENT`, `FINANCE`, `LIBRARIAN`, `IT`.

2. **SIS (Student Information)**: Admissions, enrollment, profiles, transcripts, progression, ID generation, clearance, exam card application, fees status(balance), fee payments(M-Pesa, card payments).

3. **Courses & Timetables**: Course catalog (meteorology specializations), class sections, rooms, schedules, lecturer assignment.

4. **Academic Admin**: Attendance, gradebook, exam scheduling, results publishing, GPA calculation rules.

5. **Finance**: Invoicing, receipts, fee structures, scholarships/bursaries, M-Pesa STK push integration (sandbox + prod), exports.

6. **Library**: Catalog, lending, returns, fines, integration-ready hooks for external research databases.

7. **Research & Projects**: Project registry, teams, deliverables, document repository (with safe storage).

8. **Notifications**: Email/SMS (Safaricom SMS gateway or provider-agnostic interface), in-app alerts.

9. **Reporting & Compliance**: Student performance, attendance, finance summaries, KMD/WMO compliance reports, downloadable CSV/PDF.

10. **System Admin**: User management, role/permission editor, audit logs, configuration, backups.

2) Tech Stack & Libraries

- **Backend**: Node.js 20+, Express, MySQL 8, Sequelize, Joi for validation, bcrypt, jsonwebtoken, cookie-parser, helmet, cors, express-rate-limit, multer (uploads), winston/pino for logs, uuid, dayjs, nodemailer, bull/agenda (optional jobs), swagger-ui-express / OpenAPI 3.

- **Frontend**: Next.js (App Router recommended), React Query (TanStack) for data fetching + caching, Axios, Tailwind CSS, Headless UI/Radix, React Hook Form + Zod, Context for lightweight state, recharts for charts.

- **Infra**: Docker + docker-compose, Nginx reverse proxy, PM2 (or node:18-alpine-based), GitHub Actions CI, S3-compatible object storage (e.g., MinIO) for files, Redis (jobs/cache), Mailhog in dev.

3) Target Folder Structures (Generate these)

Backend

```
backend/
├── src/
│   ├── config/
│   │   ├── env.js
│   │   ├── db.js                 # Sequelize client
│   │   └── logger.js
│   ├── middleware/
│   │   ├── auth.js               # JWT + RBAC
│   │   ├── validate.js
│   │   ├── rateLimiter.js
│   │   ├── errorHandler.js
│   │   └── cors.js
│   ├── utils/
│   │   ├── crypto.js             # bcrypt helpers
│   │   ├── jwt.js
│   │   ├── responses.js
│   │   ├── id.js                 # UUID, student IDs
│   │   └── files.js              # uploads, MIME checks
│   ├── modules/
│   │   ├── auth/
│   │   │   ├── auth.controller.js
│   │   │   ├── auth.service.js
│   │   │   ├── auth.routes.js
│   │   │   ├── auth.validation.js
│   │   │   └── auth.types.js
│   │   ├── users/ ... (same pattern)
│   │   ├── students/ ...
│   │   ├── courses/ ...
│   │   ├── academics/ ...
│   │   ├── finance/ ...
│   │   ├── library/ ...
│   │   ├── research/ ...
│   │   ├── notifications/ ...
│   │   └── reports/ ...
│   ├── jobs/
│   │   ├── index.js
│   │   └── email.scheduler.js
│   ├── routes.js                 # mount all module routes
│   ├── app.js                    # create app, middleware
│   └── server.js                 # listen, graceful shutdown
├── sequelize
│   ├── schema.prisma(sequlised)
│   └── migrations/
├── tests/ (vitest/jest + supertest)
├── .env.example
├── dockerfile
├── package.json
└── README.md
```

Frontend

```
frontend/
├── src/
│   ├── app/                      # App Router
│   │   ├── layout.jsx
│   │   ├── page.jsx              # Landing
│   │   ├── login/page.jsx
│   │   ├── dashboard/page.jsx
│   │   ├── students/page.jsx
│   │   ├── courses/page.jsx
│   │   ├── finance/page.jsx
│   │   ├── library/page.jsx
│   │   ├── research/page.jsx
│   │   ├── reports/page.jsx
│   │   └── admin/page.jsx
│   ├── components/
│   │   ├── ui/ (Navbar, Sidebar, Tabs, Modal, Toast, Loader)
│   │   ├── tables/ (DataTable w/ pagination, filters)
│   │   └── forms/ (RHForm wrappers, inputs)
│   ├── hooks/ (useAuth, useToast, useModal)
│   ├── lib/ (axios instance, auth helpers)
│   ├── styles/ (globals.css, tailwind.css)
│   ├── constants/ (roles, endpoints, tabs)
│   ├── types/ (user.d.ts, student.d.ts)
│   └── store/ (Zustand or context)
├── public/
├── tailwind.config.js
├── next.config.js
├── dockerfile
├── .env.example
└── package.json
```

4) Database Design (Generate SQL/Models + ERD)

Create a normalized schema with FKs and indexes. Minimum tables:

- `users` (id, email, password_hash, role, status, last_login_at, created_at)
- `profiles` (user_id FK, first_name, last_name, phone, gender, dob, address, national_id)
- `students` (id, user_id FK, student_no, admission_date, program_id, status)
- `lecturers` (id, user_id FK, staff_no, department_id)
- `programs` (id, name, code, level, duration_months)
- `courses` (id, program_id FK, code, title, credits, semester)
- `class_sections` (id, course_id, lecturer_id, room, schedule_json)
- `enrollments` (id, student_id, class_section_id, status)
- `attendance` (id, class_section_id, student_id, date, status)
- `assessments` (id, class_section_id, title, max_score, weight)
- `grades` (id, assessment_id, student_id, score, graded_at)
- `fee_structures` (id, program_id, item, amount_kes)
- `invoices` (id, student_id, total_kes, due_date, status)
- `invoice_items` (id, invoice_id, item, amount_kes)
- `payments` (id, invoice_id, amount_kes, method, mpesa_ref, paid_at)
- `library_items` (id, isbn, title, authors, type, copies_total, copies_available)
- `loans` (id, library_item_id, borrower_user_id, borrowed_at, due_at, returned_at, fine_kes)

- `research_projects` (id, title, lead_lecturer_id, sponsor, start_date, end_date, status)
- `project_members` (project_id, user_id, role)
- `notifications` (id, user_id, channel, title, body, sent_at, meta_json)
- `audit_logs` (id, actor_user_id, action, entity, entity_id, ip, user_agent, created_at)

Deliver: Sequelize models + migrations and an ER diagram (mermaid). Add **indexes** for lookups (email, student_no, mpesa_ref, etc.).

5) API Design (OpenAPI 3 + Controllers)

Produce an **OpenAPI spec** with these routes (indicative):

- **Auth**: `POST /auth/login`, `POST /auth/logout`, `POST /auth/refresh`, `POST /auth/forgot`, `POST /auth/reset`
- **Users/Admin**: CRUD users, assign roles, list by role
- **Students**: `POST /students` (admit), `GET /students/:id`, `GET /students?search=&page=`, `PATCH /students/:id`
- **Courses**: CRUD programs/courses/sections, timetable endpoints
- **Academics**: attendance, assessments, grading, publish results, compute GPA
- **Finance**: fee structures, invoices (`GET /students/:id/invoices`), payments (`POST /finance/mpesa/stk`, `POST /finance/mpesa/callback`)
- **Library**: catalog search, borrow/return, fines
- **Research**: CRUD projects, members, uploads
- **Notifications**: `POST /notify` (role-guarded), templates
- **Reports**: downloadable CSV/PDF endpoints
- **Admin**: audit logs, backups trigger (if applicable), config

Include **RBAC** matrix (e.g., only FINANCE can create invoices/payments; LECTURER can grade only their own sections).

6) Security Model

- **JWT**: access token (short TTL) + refresh token (longer TTL) in **httpOnly, secure, sameSite=strict** cookies. Rotate refresh tokens.
- **Password**: bcrypt with a strong cost factor, reject common passwords.
- **Validation**: Zod/Joi schemas for all inputs. Central error formatting.
- **CSRF**: Double-submit cookie or SameSite + CSRF token for state-changing endpoints from the browser.
- **CORS**: strict origins from env.
- **Rate Limiting**: per IP + per user on sensitive routes.
- **Uploads**: MIME/type/size checks, virus scan hook placeholder, store in S3/MinIO.
- **Audit**: Log sensitive actions (grades changed, payments posted, role changes).

7) Frontend (SPA UX with Next.js + Tailwind)

Deliver a **production-ready UI shell**:

- **Layouts**: AuthLayout (public), MainLayout (private) with responsive Sidebar + Navbar + Breadcrumbs + Tabs component.
- **Auth Screens**: Login (email/password), remember me, errors; optional register (admin use).
- **Dashboards**:
    - Student: timetable, enrolled courses, grades, invoices/payments, notifications.
    - Lecturer: sections, attendance taker, gradebook, submissions, messaging.
    - Admin: user management, programs, courses, reports, audit logs.

    - Finance: invoices, payments, reconciliation, exports.
    - Library: catalog, loans, fines.
- **UX**: Tables with pagination, column filters; modals for CRUD; toasts; skeleton loaders; empty states; accessible forms with RHF + Zod.
- **Theming**: Light/dark, Tailwind config.

Data layer: Axios instance with cookie credentials, React Query for caching & mutations, global auth guard (redirect to /login when unauthenticated), 401/403 handling.

8) Payments (M-Pesa)

Implement an abstraction around M-Pesa STK Push:

- **Endpoints**: `POST /finance/mpesa/stk` to initiate; `POST /finance/mpesa/callback` to confirm.
- **States**: pending → success/failed; idempotency key; store M-Pesa receipt number.
- **Testing**: sandbox credentials in `.env.example`.
- **Security**: verify callback signature/origin if available; reconcile jobs, retry on pending.

9) Notifications

- **Email**: Nodemailer (Mailhog dev), templating (MJML/Handlebars).
- **SMS**: Provider-agnostic interface; plug Safaricom or Africa’s Talking.
- **In-App**: Notification center + unread badge.

10) Docs to Generate

- `README.md` (root, backend, frontend) with:
    - Set up, env variables (document all)

    - Running dev (docker-compose), migrations, seeding
    - Testing, linting, formatting, and commit conventions
    - Deployment steps
- **OpenAPI** docs served at `/docs`.
- **Architecture diagram** (mermaid) and **ERD** (mermaid).
- **RBAC policy** table.
- **Backup/Restore** guide.

11) Dev & Prod Tooling

- **Docker Compose**: services `api`, `web`, `db` (MySQL), `redis`, `minio`, `mailhog`, `nginx`.
- **Nginx**: TLS termination, gzip, caching for static assets.
- **CI (GitHub Actions)**:
    - Lint (ESLint), type check, test (backend+frontend)
    - Build containers
    - Run migrations
    - Deploy to staging/prod (push to registry + compose up)
- **PM2**: process manager in prod, graceful shutdown, health checks.
- **Logs**: JSON logs to stdout, sample `filebeat`/ELK notes.
- **Metrics**: `/health`, `/metrics` (Prometheus format if possible).

12) Testing Strategy

- **Unit**: services, utils (vitest/jest).

- **Integration**: API with supertest and a test DB.
- **E2E**: Playwright/Cypress for critical flows (login, enroll, pay).
- **Security**: auth/permission tests; input validation tests; rate limit tests.

13) Seeding

Provide seed scripts for:

- Roles and an admin user.
- Sample programs, courses, and sections.
- Demo students/lecturers.
- Fee structures, invoices.
- Library items.

14) Make It Run

Provide all files needed to run:

1. `docker-compose.yml` (db, redis, minio, mailhog, nginx, api, web).
2. `Dockerfile` for `api` and `web` with multistage builds.
3. `Makefile` (targets: `make up`, `down`, `logs`, `migrate`, `seed`, `test`).
4. `.env.example` for both frontend and backend with documented variables (DB creds, JWT secrets, cookie settings, M-Pesa keys, SMTP, S3/MinIO, CORS origins).

15) Coding Standards

- JavaScript is preferred across both frontend and backend.
- ESLint + Prettier configs.

- Commit style: Conventional Commits.
- PR templates & CODEOWNERS examples.

16) Deliverables (Generate Now)

1. **Backend**:
   - Sequelize models + migration
   - Example controllers/services/routes for `auth`, `students`, `finance`
   - Middleware: `auth`, `rbac`, `validate`, `errorHandler`, `rateLimiter`
   - `OpenAPI.yaml` or runtime OpenAPI generator
   - Seed + migration scripts
2. **Frontend**:
   - Layouts, pages, routes
   - UI kit components (Sidebar, Navbar, Tabs, Modal, Toast, Loader)
   - Forms with RHF + Zod
   - Axios instance with cookie creds, React Query setup
   - ProtectedRoute/AuthGuard pattern
3. **Infra**:
   - `docker-compose.yml`
   - Nginx config
   - GitHub Actions CI workflow
   - PM2 ecosystem file
4. **Docs**:

  - Root `README.md` with step-by-step quickstart
  - ERD + architecture diagrams (mermaid)
  - RBAC matrix
  - Security checklist (OWASP) + Kenya DPA notes

17) Acceptance Criteria

- `docker compose up` starts full stack.
- Access **frontend** at `https://localhost` (nginx TLS self-signed ok) → login page.
- Default admin can login, create a program, course, class section.
- Student can be admitted, enrolled, marked present, graded.
- Finance can issue invoice and complete an **M-Pesa sandbox** payment flow.
- Library loan/return works; fines compute.
- Reports export CSV; notifications send to Mailhog.
- Logs are structured; `/health` returns OK; errors are standardized.
- Tests run green; migrations + seeds idempotent.

18) Theme & UI Design Guide

1. Brand Colors

We'll use **Tailwind CSS color tokens** for simplicity and maintainability.

Purpose	Color Family	Tailwind Range
Background	Slate (cool neutral)	`slate-500 → slate-950` (dark mode primary backgrounds)
Primary (CTA, Buttons, Links)	Blue	`blue-500 → blue-800`
Text (on dark bg)	White	`text-white`
Danger / Delete	Red	`red-500 → red-700`
Warning	Yellow / Orange	`yellow-400 → orange-500`
Light Mode Background	Gray / Off-White	`gray-100 → gray-200`
Light Mode Text	Slate / Dark Gray	`slate-800` / `gray-900`

---

2. Typography

- **Font Family**: Roboto (import via Google Fonts or include in Tailwind config).
- **Weights**:
  - `400` → Regular (body text)
  - `500` → Medium (labels, minor headers)
  - `700` → Bold (section headers, emphasis)

---

3. Icons

- **Primary sets**:
  - `react-icons/fa` → FontAwesome
  - `react-icons/fi` → Feather Icons
- **Guidelines**:
  - Keep icons **stroke-based (outlined)** for a clean, modern look.
  - Use **filled icons only** for status indicators (success, error).

---

4. Animations & Transitions

- **Framer Motion** (preferred) for page transitions, modals, and dashboards.
- **Tailwind transitions** for micro-interactions:
  - `transition-all ease-in-out duration-200` for hover states.

    - Button hover → background darkens (`blue-600 → blue-700`), text stays white.
    - Dropdowns, tooltips → `scale-95 → scale-100` pop-in.

---

5. Light & Dark Mode

- **Default Mode**: Dark (slate background, white text).
- **Light Mode**:
    - Background → `gray-100` / `gray-200`
    - Text → `slate-800` / `gray-900`
    - Accent colors (blue, red, yellow/orange) remain the same.
- Switch using Tailwind's `dark:` modifier or `next-themes`.

---

6. Component Guidelines

- **Buttons**:
    - Primary → `bg-blue-600 hover:bg-blue-700 text-white`
    - Danger → `bg-red-600 hover:bg-red-700 text-white`
    - Warning → `bg-orange-500 hover:bg-orange-600 text-white`
    - Rounded (`rounded-xl`), subtle shadow (`shadow-md`).
- **Cards / Containers**:
    - Dark → `bg-slate-800 text-white`

    - Light → `bg-white text-slate-800 shadow-sm`
    - Rounded corners (`rounded-2xl`), padding (`p-6`).
- **Inputs / Forms**:
    - Border → `border-slate-600 focus:border-blue-500`
    - Background → `bg-slate-900 text-white placeholder-gray-400` (dark)
    - Animated focus glow: `focus:ring-2 focus:ring-blue-500`.
- **Tables / DataGrid**:
    - Zebra stripes using `odd:bg-slate-800 even:bg-slate-700`.
    - Hover highlight → `hover:bg-slate-600`.
    - In light mode → subtle `gray-100` / `gray-50` background stripes.

---

7. Example Tailwind Config Snippet

```
// tailwind.config.js

module.exports = {

  darkMode: "class",

  theme: {

    extend: {

      fontFamily: {

        sans: ["Roboto", "sans-serif"],

      },

      colors: {
        brand: {
          bg: {
            dark: "#0f172a", // slate-950
            light: "#f8fafc", // gray-50
          },
          primary: {
            DEFAULT: "#2563eb", // blue-600
            dark: "#1e40af", // blue-800
          },
          danger: "#dc2626", // red-600
          warning: "#f59e0b", // amber-500
        },
      },
    },
  },
};
```

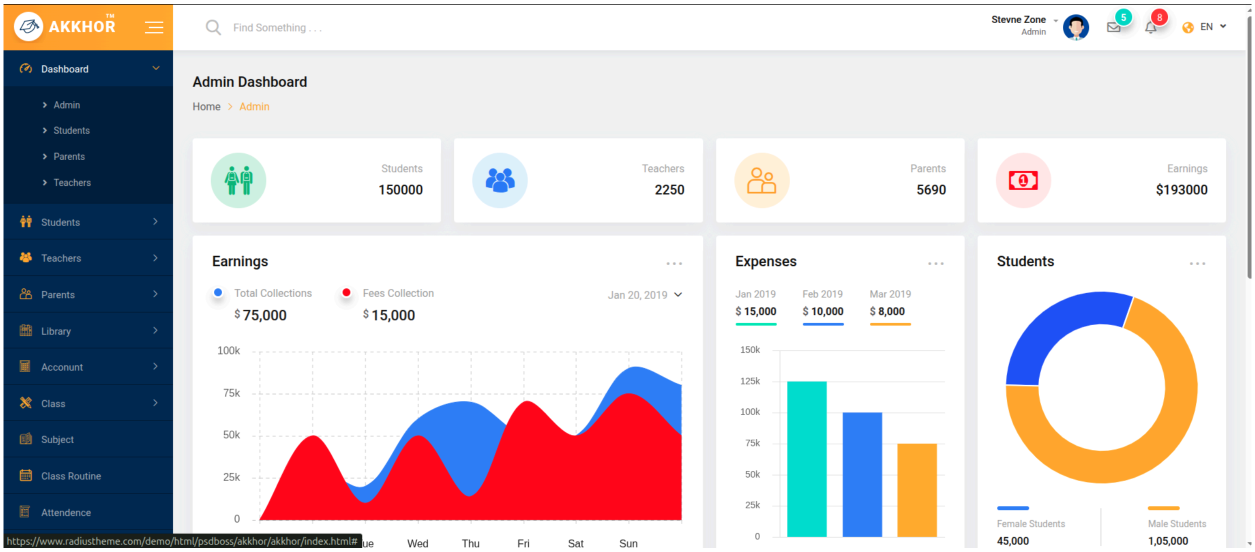
---

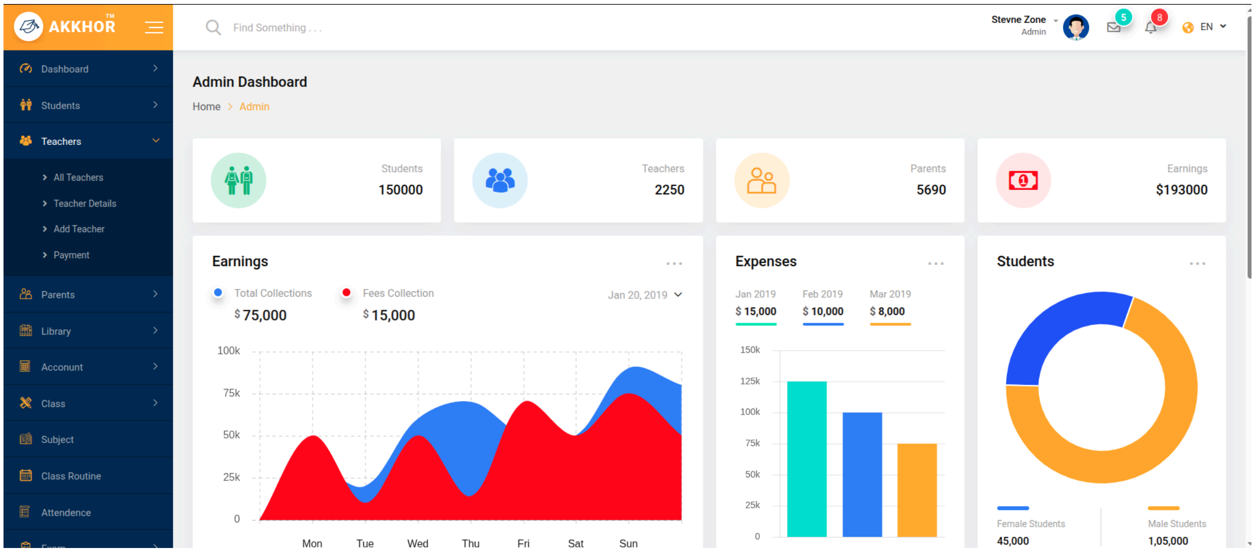
8. Deliverables

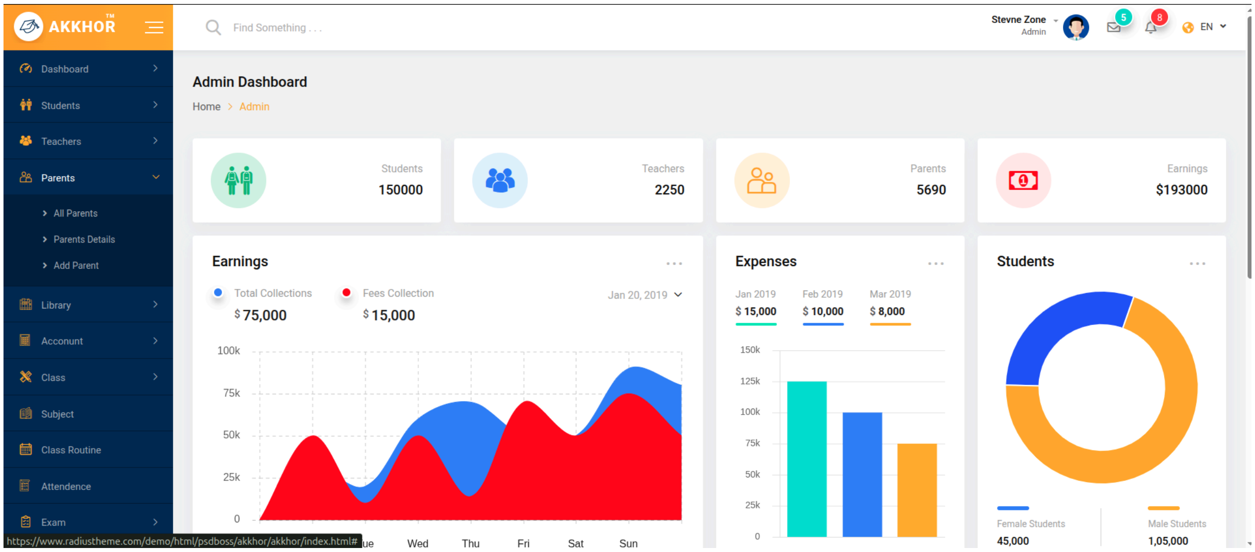
- **Figma/Storybook Design System** with:
    - Buttons, Forms, Tables, Modals, Alerts (danger/warning/success).
- **Reusable Components** in React:

    - `<Button variant="primary" />`
    - `<Card />`
    - `<Alert type="danger" />`
- **Animations**: Page fade/slide, modal pop-in/out, toast notifications.

Below are images for possible screens but not actual on the final product we are developing







AKKHOR™

Find Something . . .

Stevne Zone
Admin

5 8 EN

- Dashboard
- Students
- Teachers
- Parents
  - All Parents
  - Parents Details
  - Add Parent
- Library
- Acconunt
- Class
- Subject
- Class Routine
- Attendence
- Exam

Admin Dashboard

Home > Admin

Students	Teachers	Parents	Earnings
150000	2250	5690	$193000

Earnings

● Total Collections $ 75,000 ● Fees Collection $ 15,000 Jan 20, 2019

100k 75k 50k 25k 0

ue Wed Thu Fri Sat Sun

Expenses

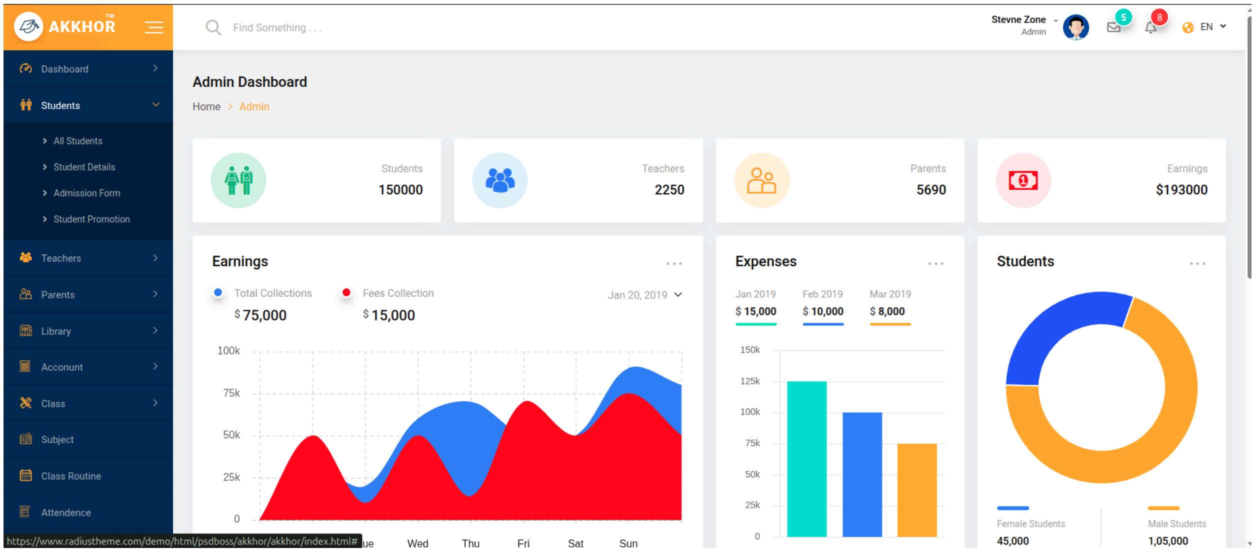
Jan 2019	Feb 2019	Mar 2019
$ 15,000	$ 10,000	$ 8,000

150k 125k 100k 75k 50k 25k 0

Students

Female Students	Male Students
45,000	1,05,000

https://www.radiustheme.com/demo/html/psdboss/akkhor/akkhor/index.html#



AKKHOR™

Find Something . . .

Stevne Zone
Admin

5 8 EN

- Dashboard
- Students
  - All Students
  - Student Details
  - Admission Form
  - Student Promotion
- Teachers
- Parents
- Library
- Acconunt
- Class
- Subject
- Class Routine
- Attendence

Admin Dashboard

Home > Admin

Students	Teachers	Parents	Earnings
150000	2250	5690	$193000

Earnings

● Total Collections $ 75,000 ● Fees Collection $ 15,000 Jan 20, 2019

100k 75k 50k 25k 0

ue Wed Thu Fri Sat Sun

Expenses

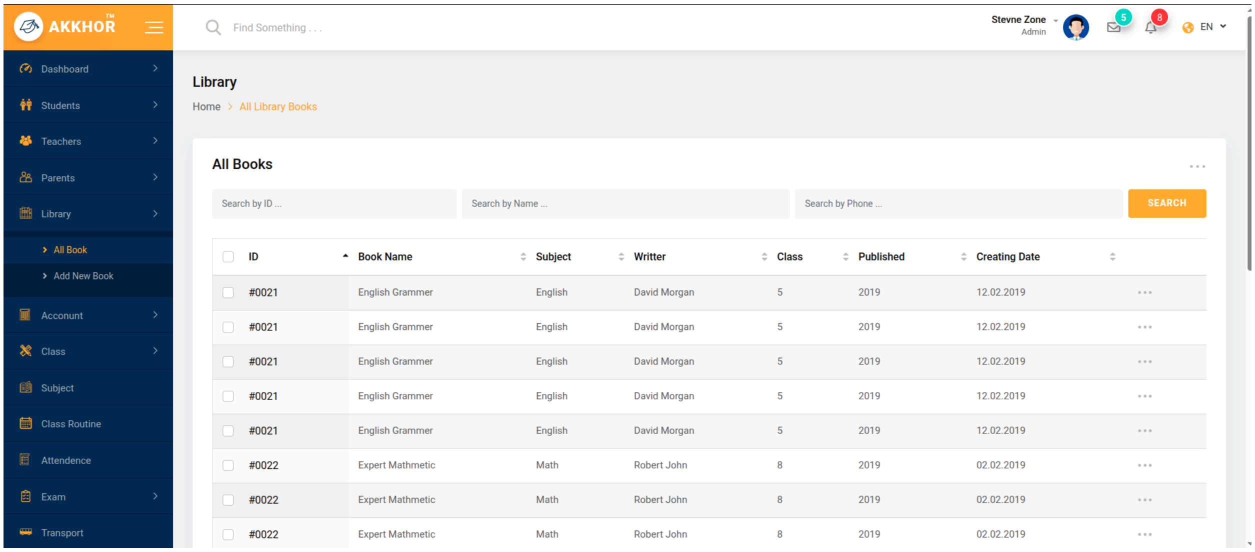
Jan 2019	Feb 2019	Mar 2019
$ 15,000	$ 10,000	$ 8,000

150k 125k 100k 75k 50k 25k 0

Students

Female Students	Male Students
45,000	1,05,000

https://www.radiustheme.com/demo/html/psdboss/akkhor/akkhor/index.html#



AKKHOR™

Find Something . . .

Stevne Zone
Admin

5 8 EN

- Dashboard
- Students
- Teachers
- Parents
- Library
  - All Book
  - Add New Book
- Acconunt
- Class
- Subject
- Class Routine
- Attendence
- Exam
- Transport

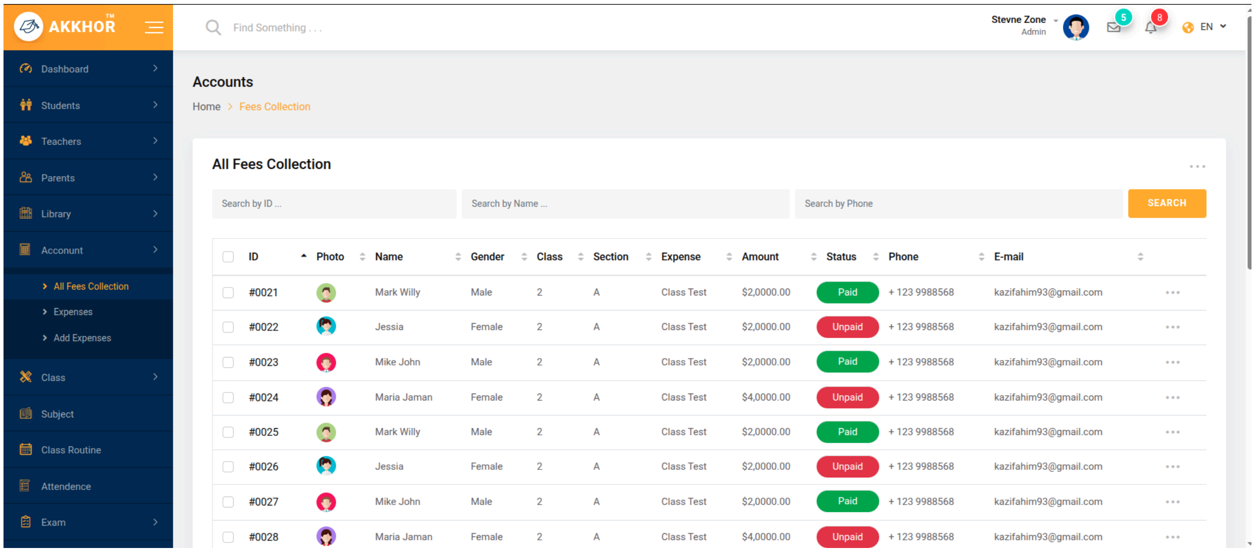
Library

Home > All Library Books

All Books

Search by ID ... | Search by Name ... | Search by Phone ... | SEARCH

□	ID	Book Name	Subject	Writter	Class	Published	Creating Date	
□	#0021	English Grammer	English	David Morgan	5	2019	12.02.2019	...
□	#0021	English Grammer	English	David Morgan	5	2019	12.02.2019	...
□	#0021	English Grammer	English	David Morgan	5	2019	12.02.2019	...
□	#0021	English Grammer	English	David Morgan	5	2019	12.02.2019	...
□	#0021	English Grammer	English	David Morgan	5	2019	12.02.2019	...
□	#0022	Expert Mathmetic	Math	Robert John	8	2019	02.02.2019	...
□	#0022	Expert Mathmetic	Math	Robert John	8	2019	02.02.2019	...
□	#0022	Expert Mathmetic	Math	Robert John	8	2019	02.02.2019	...



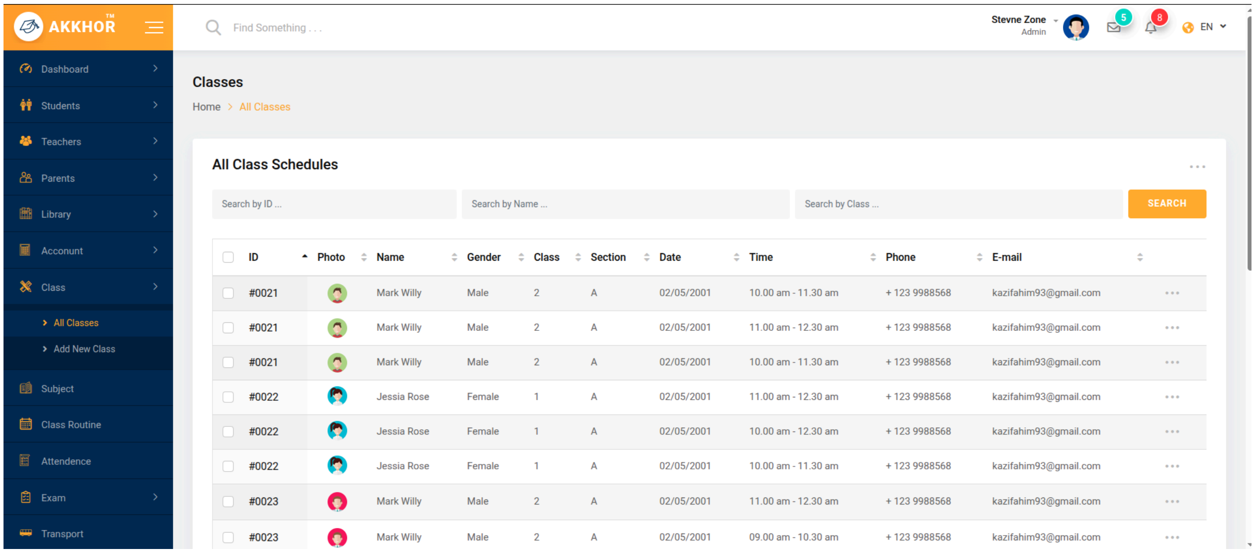
Accounts

Home > Fees Collection

All Fees Collection

Search by ID ... | Search by Name ... | Search by Phone | SEARCH

☐	ID	Photo	Name	Gender	Class	Section	Expense	Amount	Status	Phone	E-mail	
☐	#0021		Mark Willy	Male	2	A	Class Test	$2,0000.00	Paid	+ 123 9988568	kazifahim93@gmail.com	•••
☐	#0022		Jessia	Female	2	A	Class Test	$2,0000.00	Unpaid	+ 123 9988568	kazifahim93@gmail.com	•••
☐	#0023		Mike John	Male	2	A	Class Test	$2,0000.00	Paid	+ 123 9988568	kazifahim93@gmail.com	•••
☐	#0024		Maria Jaman	Female	2	A	Class Test	$4,0000.00	Unpaid	+ 123 9988568	kazifahim93@gmail.com	•••
☐	#0025		Mark Willy	Male	2	A	Class Test	$2,0000.00	Paid	+ 123 9988568	kazifahim93@gmail.com	•••
☐	#0026		Jessia	Female	2	A	Class Test	$2,0000.00	Unpaid	+ 123 9988568	kazifahim93@gmail.com	•••
☐	#0027		Mike John	Male	2	A	Class Test	$2,0000.00	Paid	+ 123 9988568	kazifahim93@gmail.com	•••
☐	#0028		Maria Jaman	Female	2	A	Class Test	$4,0000.00	Unpaid	+ 123 9988568	kazifahim93@gmail.com	•••



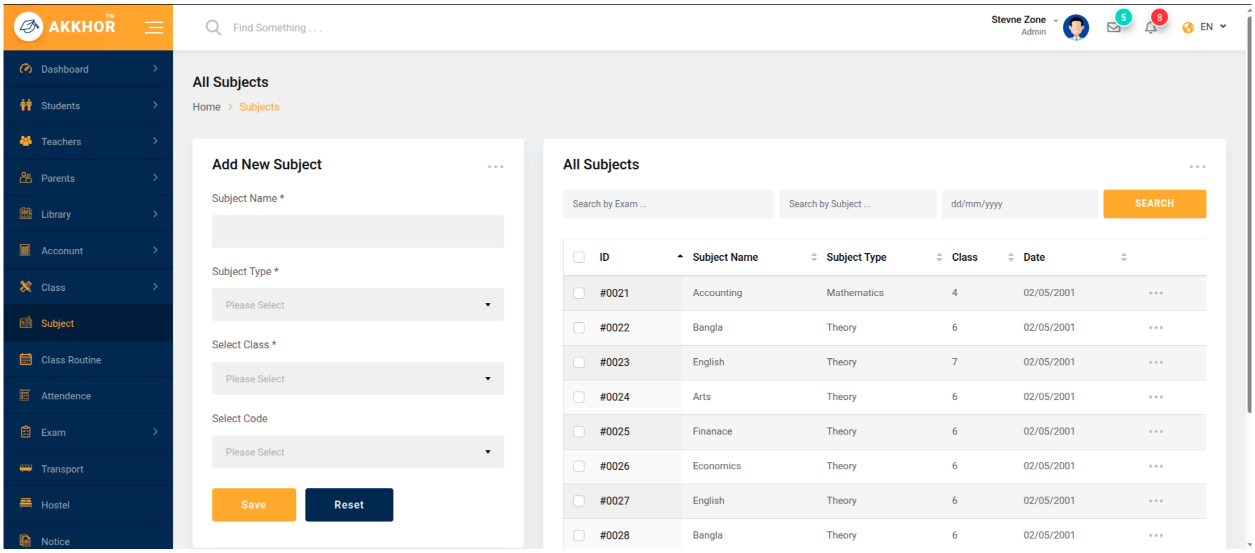
Classes

Home > All Classes

All Class Schedules

Search by ID ... | Search by Name ... | Search by Class ... | SEARCH

☐	ID	Photo	Name	Gender	Class	Section	Date	Time	Phone	E-mail	
☐	#0021		Mark Willy	Male	2	A	02/05/2001	10.00 am - 11.30 am	+ 123 9988568	kazifahim93@gmail.com	•••
☐	#0021		Mark Willy	Male	2	A	02/05/2001	11.00 am - 12.30 am	+ 123 9988568	kazifahim93@gmail.com	•••
☐	#0021		Mark Willy	Male	2	A	02/05/2001	10.00 am - 11.30 am	+ 123 9988568	kazifahim93@gmail.com	•••
☐	#0022		Jessia Rose	Female	1	A	02/05/2001	11.00 am - 12.30 am	+ 123 9988568	kazifahim93@gmail.com	•••
☐	#0022		Jessia Rose	Female	1	A	02/05/2001	10.00 am - 12.30 am	+ 123 9988568	kazifahim93@gmail.com	•••
☐	#0022		Jessia Rose	Female	1	A	02/05/2001	10.00 am - 11.30 am	+ 123 9988568	kazifahim93@gmail.com	•••
☐	#0023		Mark Willy	Male	2	A	02/05/2001	11.00 am - 12.30 am	+ 123 9988568	kazifahim93@gmail.com	•••
☐	#0023		Mark Willy	Male	2	A	02/05/2001	09.00 am - 10.30 am	+ 123 9988568	kazifahim93@gmail.com	•••



All Subjects

Home > Subjects

Add New Subject

Subject Name *

Subject Type *

Please Select

Select Class *

Please Select

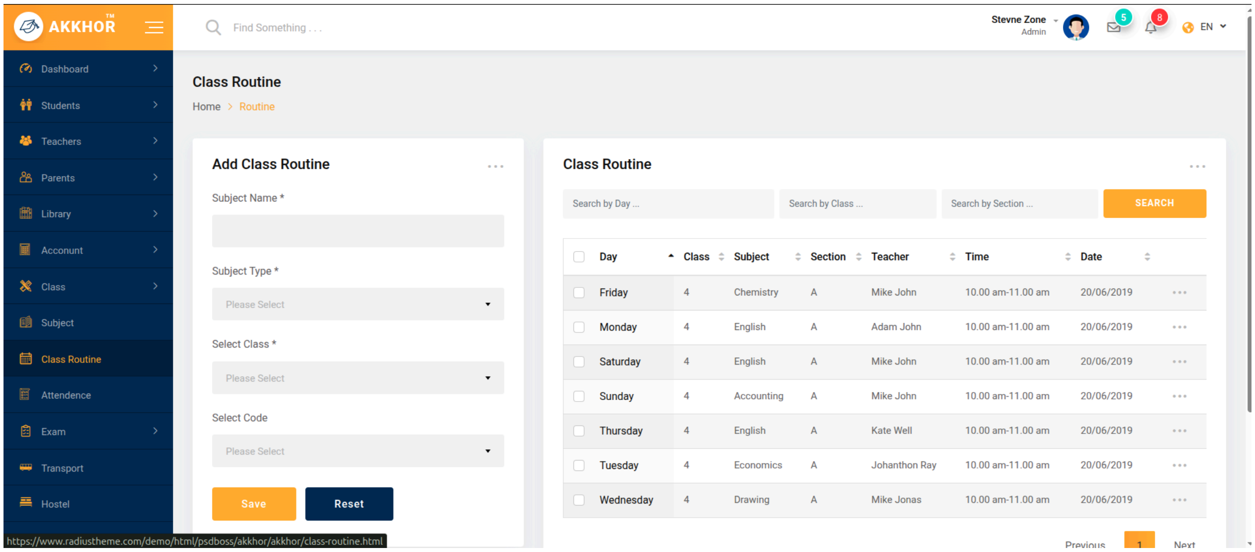
Select Code

Please Select

All Subjects

Search by Exam ... | Search by Subject ... | dd/mm/yyyy | SEARCH

☐	ID	Subject Name	Subject Type	Class	Date	
☐	#0021	Accounting	Mathematics	4	02/05/2001	•••
☐	#0022	Bangla	Theory	6	02/05/2001	•••
☐	#0023	English	Theory	7	02/05/2001	•••
☐	#0024	Arts	Theory	6	02/05/2001	•••
☐	#0025	Finanace	Theory	6	02/05/2001	•••
☐	#0026	Economics	Theory	6	02/05/2001	•••
☐	#0027	English	Theory	6	02/05/2001	•••
☐	#0028	Bangla	Theory	6	02/05/2001	•••



Class Routine

Home > Routine

Add Class Routine

Subject Name *

Subject Type *

Please Select

Select Class *

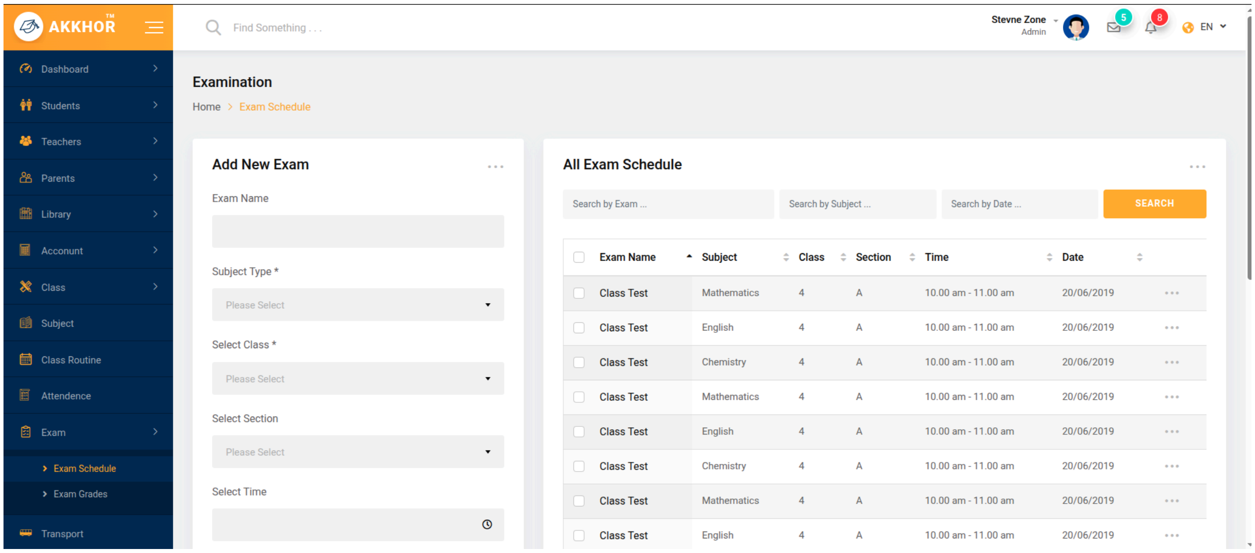
Please Select

Select Code

Please Select

Class Routine

□	Day	Class	Subject	Section	Teacher	Time	Date	
□	Friday	4	Chemistry	A	Mike John	10.00 am-11.00 am	20/06/2019	•••
□	Monday	4	English	A	Adam John	10.00 am-11.00 am	20/06/2019	•••
□	Saturday	4	English	A	Mike John	10.00 am-11.00 am	20/06/2019	•••
□	Sunday	4	Accounting	A	Mike John	10.00 am-11.00 am	20/06/2019	•••
□	Thursday	4	English	A	Kate Well	10.00 am-11.00 am	20/06/2019	•••
□	Tuesday	4	Economics	A	Johanthon Ray	10.00 am-11.00 am	20/06/2019	•••
□	Wednesday	4	Drawing	A	Mike Jonas	10.00 am-11.00 am	20/06/2019	•••



Examination

Home > Exam Schedule

Add New Exam

Exam Name

Subject Type *

Please Select

Select Class *

Please Select

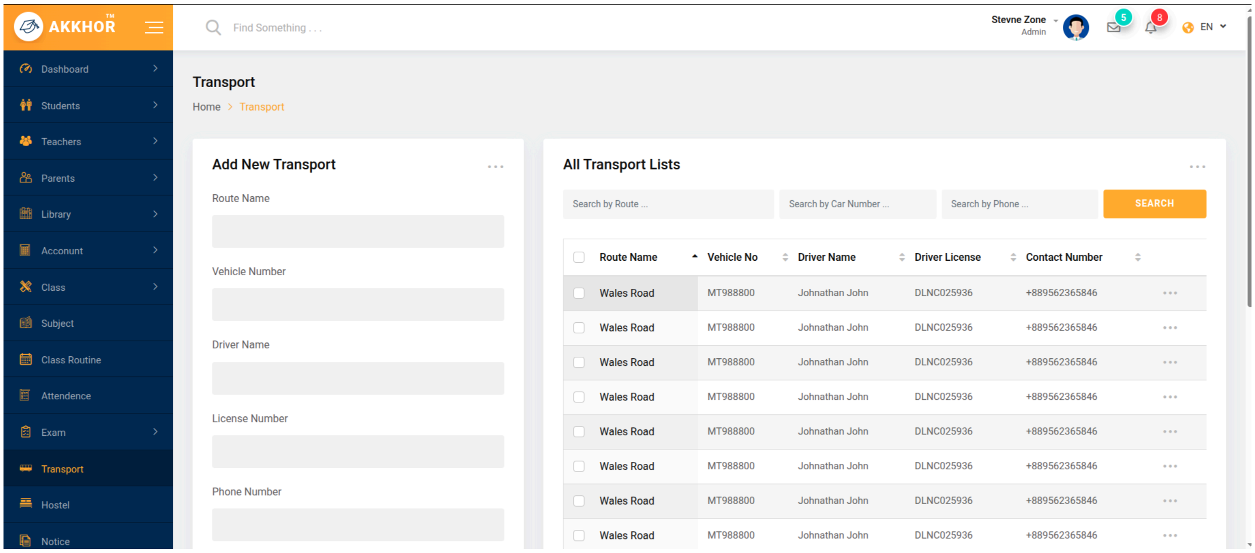
Select Section

Please Select

Select Time

All Exam Schedule

□	Exam Name	Subject	Class	Section	Time	Date	
□	Class Test	Mathematics	4	A	10.00 am - 11.00 am	20/06/2019	•••
□	Class Test	English	4	A	10.00 am - 11.00 am	20/06/2019	•••
□	Class Test	Chemistry	4	A	10.00 am - 11.00 am	20/06/2019	•••
□	Class Test	Mathematics	4	A	10.00 am - 11.00 am	20/06/2019	•••
□	Class Test	English	4	A	10.00 am - 11.00 am	20/06/2019	•••
□	Class Test	Chemistry	4	A	10.00 am - 11.00 am	20/06/2019	•••
□	Class Test	Mathematics	4	A	10.00 am - 11.00 am	20/06/2019	•••
□	Class Test	English	4	A	10.00 am - 11.00 am	20/06/2019	•••



Transport

Home > Transport

Add New Transport

Route Name

Vehicle Number

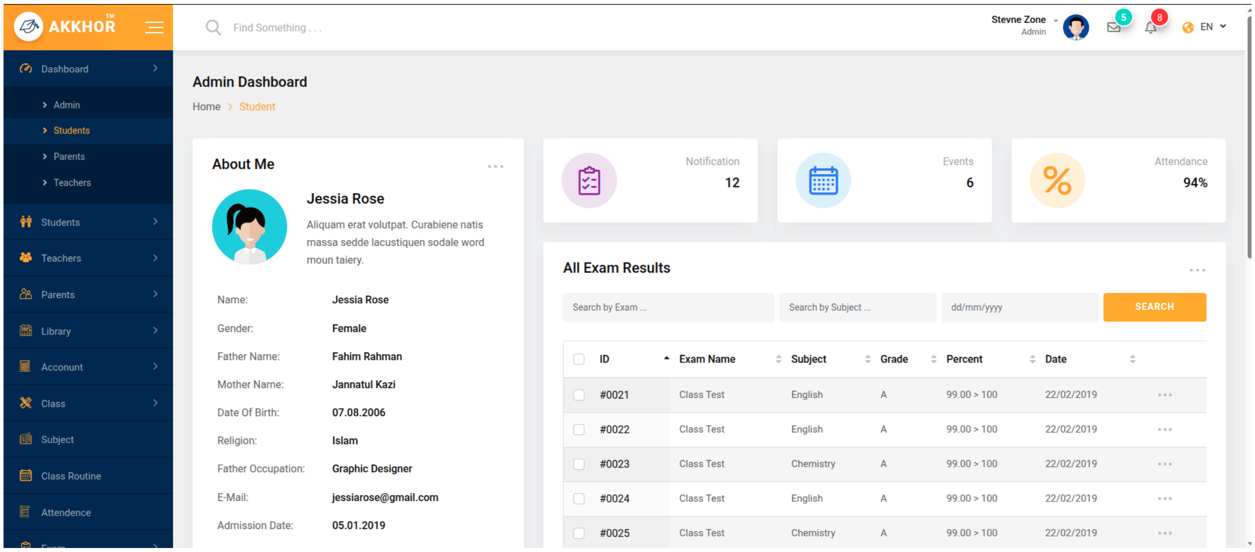
Driver Name

License Number

Phone Number

All Transport Lists

□	Route Name	Vehicle No	Driver Name	Driver License	Contact Number	
□	Wales Road	MT988800	Johnathan John	DLNC025936	+889562365846	•••
□	Wales Road	MT988800	Johnathan John	DLNC025936	+889562365846	•••
□	Wales Road	MT988800	Johnathan John	DLNC025936	+889562365846	•••
□	Wales Road	MT988800	Johnathan John	DLNC025936	+889562365846	•••
□	Wales Road	MT988800	Johnathan John	DLNC025936	+889562365846	•••
□	Wales Road	MT988800	Johnathan John	DLNC025936	+889562365846	•••
□	Wales Road	MT988800	Johnathan John	DLNC025936	+889562365846	•••
□	Wales Road	MT988800	Johnathan John	DLNC025936	+889562365846	•••



Admin Dashboard

Home > Student

About Me

Jessia Rose

Aliquam erat volutpat. Curabiene natis massa sedde lacustiquen sodale word moun taiery.

Name:	Jessia Rose
Gender:	Female
Father Name:	Fahim Rahman
Mother Name:	Jannatul Kazi
Date Of Birth:	07.08.2006
Religion:	Islam
Father Occupation:	Graphic Designer
E-Mail:	jessiarose@gmail.com
Admission Date:	05.01.2019

Notification 12

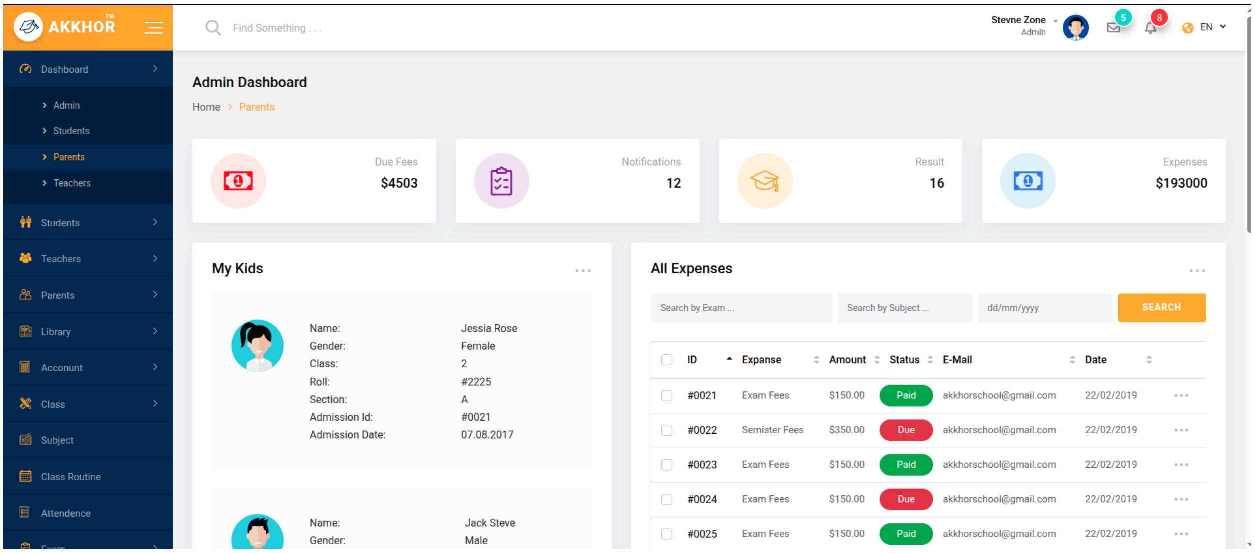
Events 6

Attendance 94%

All Exam Results

Search by Exam ... | Search by Subject ... | dd/mm/yyyy | SEARCH

□	ID	Exam Name	Subject	Grade	Percent	Date
□	#0021	Class Test	English	A	99.00 > 100	22/02/2019
□	#0022	Class Test	English	A	99.00 > 100	22/02/2019
□	#0023	Class Test	Chemistry	A	99.00 > 100	22/02/2019
□	#0024	Class Test	English	A	99.00 > 100	22/02/2019
□	#0025	Class Test	Chemistry	A	99.00 > 100	22/02/2019



Admin Dashboard

Home > Parents

Due Fees $4503

Notifications 12

Result 16

Expenses $193000

My Kids

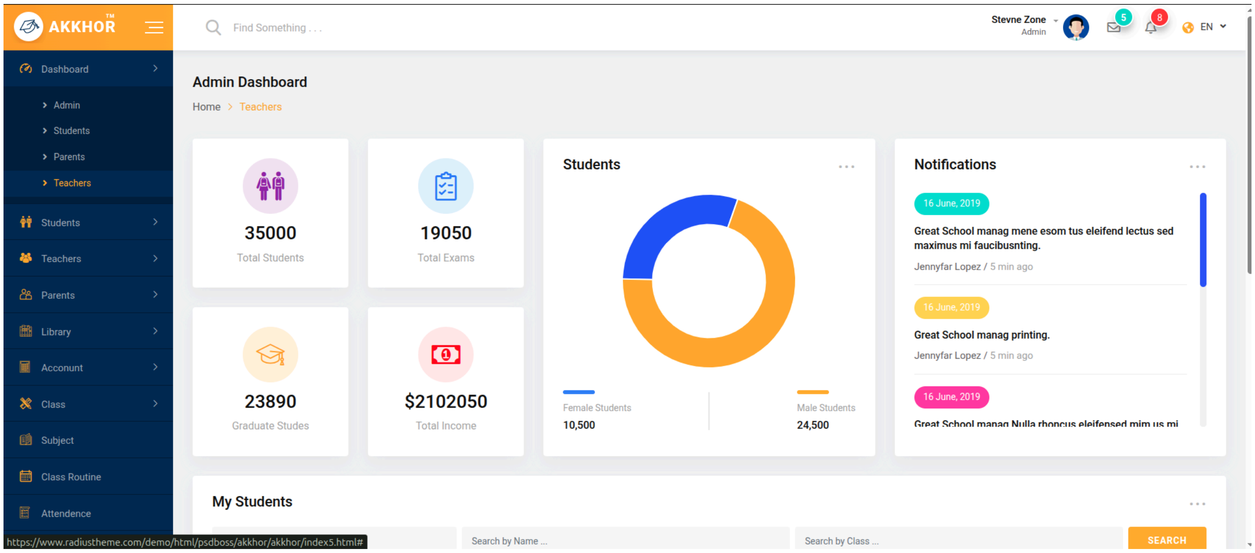
Name:	Jessia Rose
Gender:	Female
Class:	2
Roll:	#2225
Section:	A
Admission Id:	#0021
Admission Date:	07.08.2017

Name:	Jack Steve
Gender:	Male

All Expenses

Search by Exam ... | Search by Subject ... | dd/mm/yyyy | SEARCH

□	ID	Expanse	Amount	Status	E-Mail	Date
□	#0021	Exam Fees	$150.00	Paid	akkhorschool@gmail.com	22/02/2019
□	#0022	Semister Fees	$350.00	Due	akkhorschool@gmail.com	22/02/2019
□	#0023	Exam Fees	$150.00	Paid	akkhorschool@gmail.com	22/02/2019
□	#0024	Exam Fees	$150.00	Due	akkhorschool@gmail.com	22/02/2019
□	#0025	Exam Fees	$150.00	Paid	akkhorschool@gmail.com	22/02/2019



Admin Dashboard

Home > Teachers

35000 Total Students

19050 Total Exams

23890 Graduate Studes

$2102050 Total Income

Students

Female Students 10,500

Male Students 24,500

Notifications

16 June, 2019

Great School manag mene esom tus eleifend lectus sed maximus mi faucibusnting.

Jennyfar Lopez / 5 min ago

16 June, 2019

Great School manag printing.

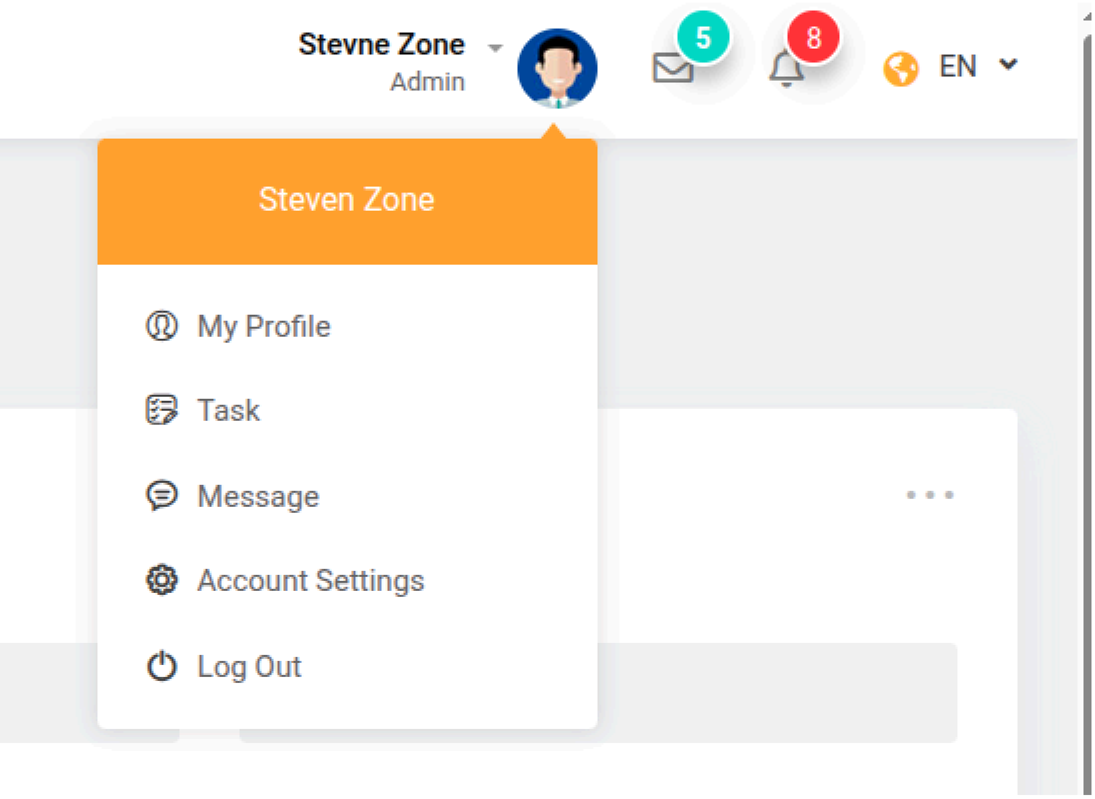
Jennyfar Lopez / 5 min ago

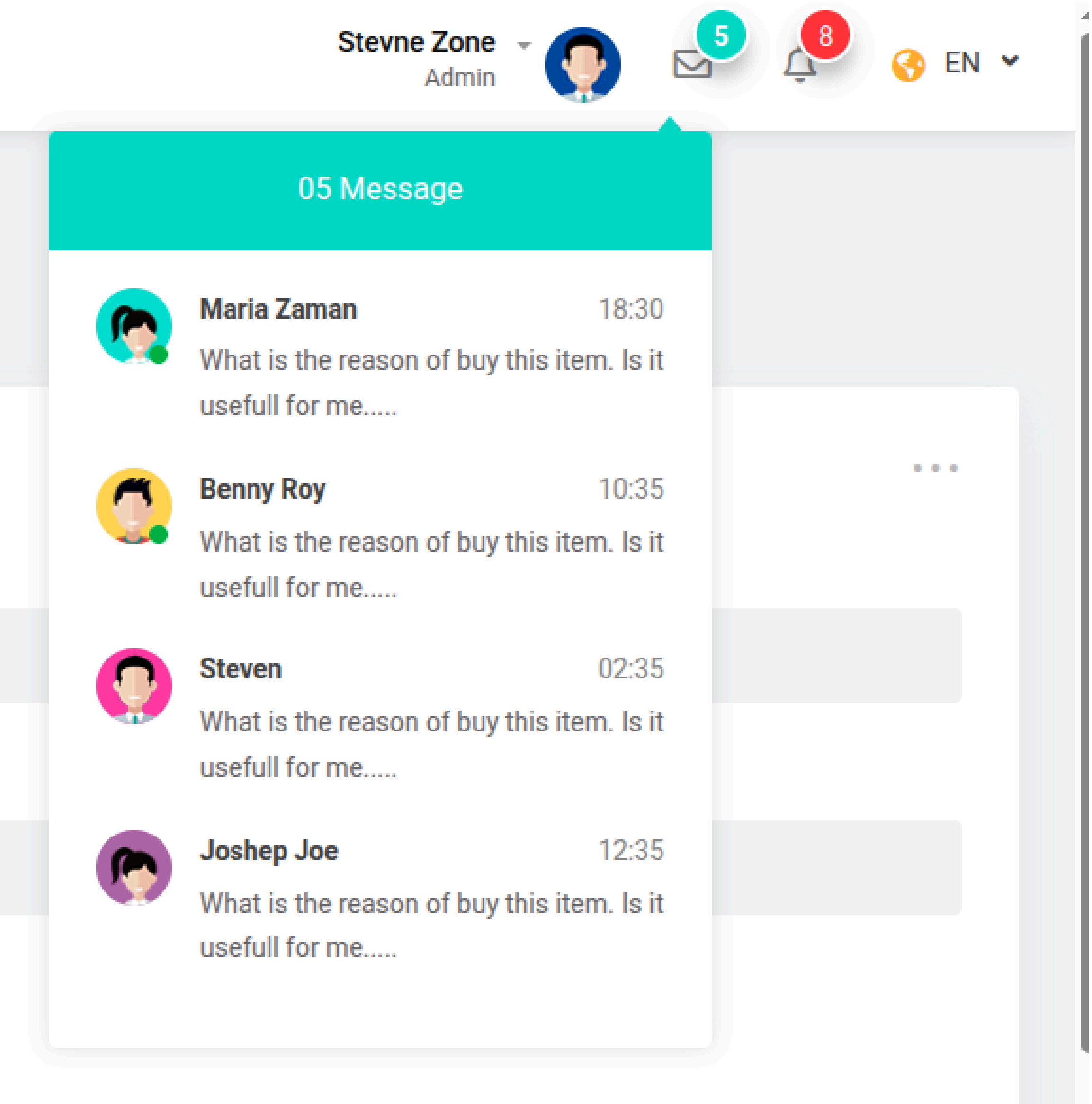
16 June, 2019

Great School manag Nulla rhoncus eleifensed mim us mi

My Students

Search by Name ... | Search by Class ... | SEARCH


Stevne Zone
Admin
5
8
EN
Steven Zone
My Profile
Task
Message
Account Settings
Log Out



05 Message

Maria Zaman 18:30

What is the reason of buy this item. Is it usefull for me.....

Benny Roy 10:35

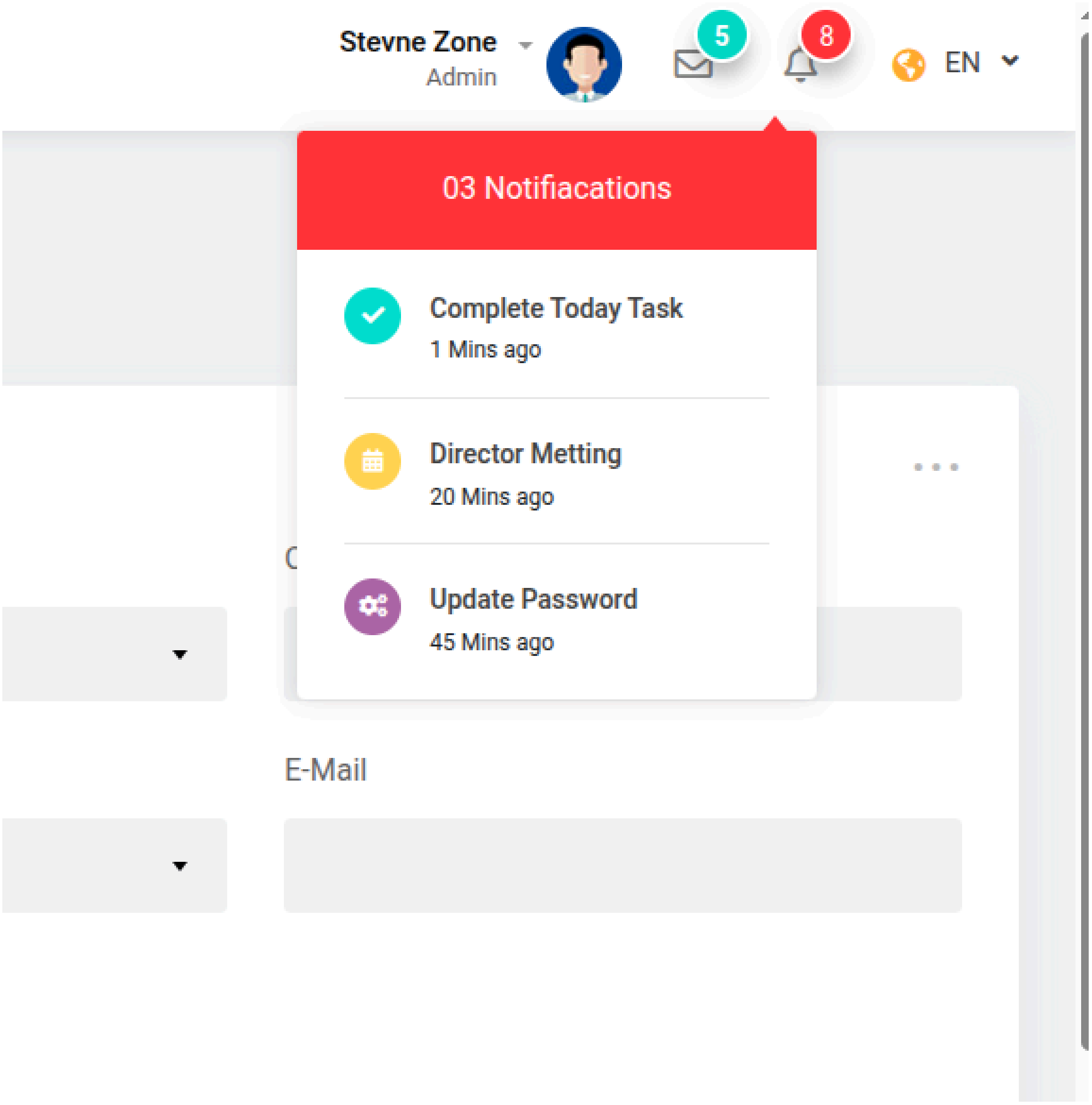
What is the reason of buy this item. Is it usefull for me.....

Steven 02:35

What is the reason of buy this item. Is it usefull for me.....

Joshep Joe 12:35

What is the reason of buy this item. Is it usefull for me.....

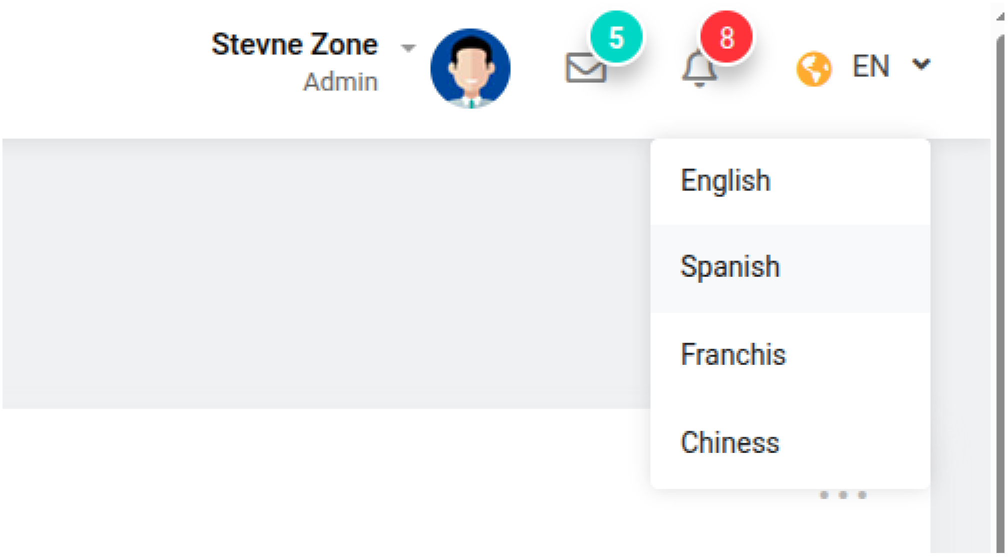


03 Notifiacations

Complete Today Task
1 Mins ago

Director Metting
20 Mins ago

Update Password
45 Mins ago


Stevne Zone
Admin
5
8
EN
English
Spanish
Franchis
Chiness