

Tasbiaat & Mamolaat Web App - Action Plan

■■■■■ ■■ ■■■■■■ (Levels of Saalik in Urdu):

- **2000 年 12 月 31 日**
- **2001 年 1 月 1 日**
- **2001 年 1 月 2 日**
- **2001 年 1 月 3 日**
- **2001 年 1 月 4 日**
- **2001 年 1 月 5 日**

Phase A — Project scaffolding & infra

- Setup repositories for frontend (Next.js) and backend (Flask).
- Provision MongoDB Atlas cluster and secure environment variables.
- Setup CI pipeline with linting and tests.

Phase B — Backend core: models, auth, REST API

- Implement Flask project structure with routes for auth, saalik, murabi, reports.
- JWT-based authentication with role-based authorization.
- REST APIs for mobile/web compatibility.

Phase C — Database design

- Collections: users, entries, levels, reports.
- Indexes on users.role, entries.date, users.region.

Phase D — Frontend scaffolding

- Setup Next.js with TypeScript, React Query, TailwindCSS.
- Responsive design with role-based dashboards.

Phase E — Implement features

- Saalik daily entries with conditional fields based on level.
- Murabi review/comments and Saalik level updates.
- Masool weekly summary aggregation.
- Sheikh monthly report with filters.
- Admin: user and region management.

Phase F — Testing & QA

- Backend unit tests, frontend component tests, E2E tests with Cypress/Playwright.
- Accessibility checks with axe.

Phase G — Deployment & Security

- Containerize backend with Docker, deploy frontend on Vercel/Netlify.
- Secure DB with IP allowlists and roles.
- Enable HTTPS, rate limiting, error monitoring.

Acceptance Criteria:

- Role-based authentication & routing enforced.
- Saalik entries conditional on levels, validated and stored.
- Murabi review and level update flows working.
- Masool weekly and Sheikh monthly reports with filters & exports.
- Responsive UI on desktop & mobile.
- Notifications implemented for daily reminders.