# In-Charge E3 — Admin Dashboard

## Squad E3.1 — User Management (view, suspend, restore)

1. Purpose  
Enable administrators to manage user accounts by viewing profiles, suspending, and restoring access.

2. Scope (MVP)  
Include: user list view, suspend/restore actions, activity logging.

3. Functional Requirements  
- View all users with search/filter.  
- Suspend user account (sets status to suspended).  
- Restore suspended account.  
- Log all admin actions.

4. Data Models  
users (id, name, email, status, created\_at, updated\_at)  
admin\_actions (id, admin\_id, action, target\_user\_id, created\_at)

5. API List  
GET /api/v1/admin/users  
POST /api/v1/admin/users/{id}/suspend  
POST /api/v1/admin/users/{id}/restore

6. Sequence Flow  
Admin → view users → select suspend/restore → system updates status → log action.

7. Testing Plan  
Unit: suspend/restore logic.  
Integration: view + suspend + restore flow.  
Edge: already suspended/restored users.

8. Deliverables  
Endpoints, DB migrations, Swagger docs, tests.

9. 8-Week Timeline  
1: schema design  
2–3: list & search endpoint  
4: suspend/restore endpoints  
5: logging integration  
6: security review  
7: tests  
8: docs & demo

## Squad E3.2 — Marketplace Oversight (flagged listings, disputes)

1. Purpose  
Allow administrators to oversee marketplace listings and disputes flagged by users.

2. Scope (MVP)  
Include: view flagged listings, mark as resolved, manage disputes.

3. Functional Requirements  
- Retrieve flagged listings.  
- Mark listing as reviewed/resolved.  
- Track user disputes with resolution status.

4. Data Models  
flagged\_listings (id, listing\_id, reason, status, created\_at)  
disputes (id, user\_id, listing\_id, status, created\_at, resolved\_at)

5. API List  
GET /api/v1/admin/marketplace/flagged  
POST /api/v1/admin/marketplace/disputes/{id}/resolve

6. Sequence Flow  
Admin → view flagged items → resolve → update DB → notify involved parties.

7. Testing Plan  
Unit: resolve logic.  
Integration: flagged listing review flow.  
Negative: invalid listing/dispute ID.

8. Deliverables  
APIs, DB schema, tests, documentation.

9. 8-Week Timeline  
1: schema setup  
2–3: flagged listing API  
4: disputes handling  
5: notifications  
6: integration testing  
7: performance checks  
8: final demo

## Squad E3.3 — Services Oversight (flagged bookings, drivers, providers)

1. Purpose  
Provide oversight for flagged service bookings and service providers (drivers, vendors).

2. Scope (MVP)  
Include: view flagged bookings, suspend/restore providers.

3. Functional Requirements  
- Retrieve flagged bookings.  
- Suspend provider account.  
- Restore provider account.  
- Log all actions.

4. Data Models  
flagged\_bookings (id, booking\_id, reason, status, created\_at)  
providers (id, name, status, created\_at, updated\_at)  
admin\_actions (extends user mgmt model)

5. API List  
GET /api/v1/admin/services/flagged  
POST /api/v1/admin/providers/{id}/suspend  
POST /api/v1/admin/providers/{id}/restore

6. Sequence Flow  
Admin → review flagged booking → suspend/restore provider → log action.

7. Testing Plan  
Unit: suspend/restore provider.  
Integration: flagged booking resolution.  
Edge: provider already suspended.

8. Deliverables  
Endpoints, DB models, logs, documentation, tests.

9. 8-Week Timeline  
1: schema finalize  
2–3: flagged bookings API  
4: suspend/restore endpoints  
5: action logging  
6: integration tests  
7: security hardening  
8: demo & docs

## Squad E3.4 — Payments Oversight (disputes, flagged transactions)

1. Purpose  
Monitor and resolve payment disputes and flagged suspicious transactions.

2. Scope (MVP)  
Include: view flagged transactions, resolve disputes, escalate if needed.

3. Functional Requirements  
- Retrieve flagged transactions.  
- Resolve or escalate disputes.  
- Maintain audit logs.

4. Data Models  
flagged\_transactions (id, txn\_id, reason, status, created\_at)  
payment\_disputes (id, txn\_id, user\_id, status, created\_at, resolved\_at)  
audit\_logs (id, admin\_id, action, details, created\_at)

5. API List  
GET /api/v1/admin/payments/flagged  
POST /api/v1/admin/payments/disputes/{id}/resolve

6. Sequence Flow  
Admin → view flagged txn → resolve dispute → update DB → notify users → log action.

7. Testing Plan  
Unit: dispute resolution.  
Integration: flagged txn workflow.  
Negative: invalid txn/dispute ID.

8. Deliverables  
Endpoints, migrations, logs, documentation, tests.

9. 8-Week Timeline  
1: schema design  
2–3: flagged txn API  
4: dispute resolution  
5: escalation flow  
6: test coverage  
7: performance tests  
8: docs & demo

## Squad E3.5 — Dashboard Analytics (basic charts, logs)

1. Purpose  
Provide analytics dashboard with key metrics, charts, and admin action logs.

2. Scope (MVP)  
Include: basic metrics visualization, logs of admin actions.

3. Functional Requirements  
- Generate charts for user activity, disputes, flagged items.  
- Provide downloadable logs.  
- Support filtering by date range.

4. Data Models  
analytics\_logs (id, metric, value, recorded\_at)  
admin\_actions (reused from previous squads)

5. API List  
GET /api/v1/admin/analytics/overview  
GET /api/v1/admin/analytics/logs

6. Sequence Flow  
Admin → access dashboard → view charts/logs → apply filters → export logs.

7. Testing Plan  
Unit: chart data queries.  
Integration: dashboard load tests.  
Edge: empty data sets.

8. Deliverables  
Analytics endpoints, dashboard UI, export functionality, tests, documentation.

9. 8-Week Timeline  
1: metrics definition  
2–3: backend endpoints  
4: charting integration  
5: logs export  
6: performance checks  
7: tests  
8: docs & demo