

# Shaheen Insurance Backend Documentation

## 1. Project Overview

This backend powers the Shaheen Insurance mobile application. It provides APIs for:

- User authentication (register/login + OTP flows)
- Motor insurance: premium calculation, proposal submission, image uploads metadata
- Travel insurance: premium calculation, proposal submission, multi-destination + student validation
- Payments: real integration (gateway) + payment record tracking
- My Policies / My Claims: mock-core endpoints (cache tables)
- Backoffice Admin Dashboard APIs: summary + listing views (motor, travel, payments) with filters/search/pagination
- Notifications, FAQs, Support Requests

## Tech Stack

- Node.js (JavaScript, not TypeScript)
  - Express.js
  - MySQL (InnoDB)
  - mysql2 (promise queries)
  - JWT authentication
- 

## 2. Folder Structure

Typical structure used in this project:

backend/

  src/

    app.js

    server.js

  config/

```
env.js  
db/  
  db.js  
middleware/  
  auth.js  
  adminOnly.js  
  validate.js  
  errorHandler.js  
utils/  
  response.js  
  paginate.js  
modules/  
  auth/  
    auth.routes.js  
    auth.controller.js  
    auth.service.js  
    auth.validation.js  
  motor/  
    motor.routes.js  
    motor.controller.js  
    motor.service.js  
    motor.validation.js  
travel/  
  travel.routes.js  
  travel.controller.js  
  travel.service.js
```

```
travel.validation.js  
payments/  
  payments.routes.js  
  payments.controller.js  
  payments.service.js  
  payments.validation.js  
admin/  
  admin.routes.js  
  admin.controller.js  
misc/  
  faqs.routes.js  
  faqs.controller.js  
  notifications.routes.js  
  support.routes.js  
sql/  
  schema.sql  
  seed.sql  
  seed_big.sql  
package.json  
.env.example
```

---

### **3. Environment Setup**

#### **3.1 Requirements**

- Node.js 18+ recommended
- MySQL 8+
- npm or yarn

### **3.2 Install & Run**

npm install

npm run dev

or

npm start

### **3.3 Environment Variables**

Create .env from .env.example.

Example:

PORT=5000

NODE\_ENV=development

DB\_HOST=localhost

DB\_PORT=3306

DB\_USER=root

DB\_PASS=your\_password

DB\_NAME=shaheen\_app

JWT\_SECRET=super\_secret\_key

JWT\_EXPIRES\_IN=7d

PAYMENT\_GATEWAY=PayFast

PAYFAST\_MERCHANT\_ID=xxxx

PAYFAST\_MERCHANT\_KEY=xxxx

PAYFAST\_RETURN\_URL=https://yourapp.com/payment/success

PAYFAST\_CANCEL\_URL=https://yourapp.com/payment/cancel

PAYFAST\_NOTIFY\_URL=https://yourapi.com/api/payments/webhook

---

## **4. Database**

### **4.1 Schema**

Database contains these groups:

#### **Lookup Tables**

- cities
- vehicle\_makes
- vehicle\_submakes
- tracker\_companies
- travel\_destinations

#### **Core**

- users
- otp\_codes

#### **Insurance**

- motor\_proposals
- motor\_vehicle\_images
- travel\_proposals
- travel\_destinations\_selected

#### **Payments**

- payments

#### **Support / CMS**

- notifications
- faqs
- support\_requests

#### **Cache (Mock Core)**

- policies\_cache

- claims\_cache

Foreign keys are enforced (InnoDB) and require correct insert order during seeding.

## 4.2 Seeding

Two seed scripts exist:

- seed.sql → small seed for basic testing
- seed\_big.sql → big seed (~50+ records across tables) for pagination/search testing

Run:

```
mysql -u root -p shaheen_app < src/sql/seed_big.sql
```

---

## 5. Authentication

### 5.1 JWT Auth

All protected endpoints require:

Authorization: Bearer <token>

JWT payload includes:

- userId
- role

### 5.2 Roles

- customer → mobile app user
- admin → backoffice dashboard user

Admin routes additionally require adminOnly middleware.

---

## 6. API Conventions

### 6.1 Base URL

<http://localhost:4000/>

### 6.2 Response Format

Success:

```
{  
  "success": true,  
  "message": "OK",  
  "data": {}  
}
```

Error:

```
{  
  "success": false,  
  "message": "Validation error",  
  "errors": [  
    { "field": "mobile", "message": "Mobile is required" }  
  ]  
}
```

---

## 7. Modules & Endpoints

### 7.1 Auth Module

#### Register

POST /api/auth/register

#### Body

```
{  
  "fullName": "Ali Khan",  
  "email": "ali@example.com",  
  "mobile": "03009998888",  
  "password": "Test@1234",
```

}

## **Login**

POST /api/auth/login

### **Body**

```
{  
  "mobile": "03009998888",  
  "password": "Test@1234"  
}
```

Returns JWT.

## **Forgot Password (OTP)**

POST /api/auth/forgot-password/otp

### **Body**

```
{ "mobile": "03009998888" }
```

Creates OTP in otp\_codes.

## **Verify OTP**

POST /api/auth/forgot-password/verify

### **Body**

```
{ "mobile": "03009998888", "otp": "123456", "newPassword": "Test1234@5678" }
```

---

## **7.2 Motor Module**

### **Calculate Premium**

POST /api/motor/calculate-premium

**Purpose**

Calculate motor premium and sum insured

Very simple sample logic:

- sumInsured = vehicleValue
- baseRate = 2% of sumInsured
- if tracker: 10% discount
- if vehicle age > 5 years: +15% loading

**Body (example)**

```
{  
    "product_type": "private",  
    "make_id": 1,  
    "submake_id": 1,  
    "model_year": 2020,  
    "sum_insured": 2500000,  
    "tracker_company_id": 1,  
    "accessories_value": 50000  
}
```

**Market Value**

POST /api/motor/market-value

**Purpose**

Returns a computed/lookup market value based on vehicle info.

Assume base = 3,000,000 for new car and depreciate 8% per year

**Body (example)**

```
{  
    "makeld": 1,  
    "submakeld": 3,
```

```
"year": 2019
```

```
}
```

## Submit Proposal

POST /api/motor/submit-proposal

Creates a row in motor\_proposals and optional images in motor\_vehicle\_images.

### form-data (example)

KEY: personalDetails

KEY-format: text

VALUE: {

```
"name": "Ali Khan",
```

```
"address": "Gulshan-e-Iqbal Block 5",
```

```
"cityId": 1,
```

```
"cnic": "42101-1234567-1",
```

```
"cnicExpiry": "2030-12-31",
```

```
"dob": "1995-05-10",
```

```
"nationality": "Pakistani",
```

```
"gender": "male"
```

```
}
```

KEY: vehicleDetails

KEY-format: text

VALUE: {

```
"productType": "private",
```

```
"registrationNumber": "ABC-123",
```

```
"appliedFor": true,
```

```
"engineNumber": "ENG1234567",
```

```
"chassisNumber": "CHS1234567",
```

```
"makeId": 1,  
"submakeId": 3,  
"modelYear": 2021,  
"colour": "White",  
"trackerCompanyId": 2,  
"accessoriesValue": 50000,  
"vehicleValue": 2800000  
}
```

KEY: images

KEY-format: File

VALUE: upload car image(s)

---

### **7.3 Travel Module**

#### **Calculate Premium**

POST /api/travel/calculate-premium

Calculate Travel premium according to FRD-style rules:

- basePremium = ratePerDay \* tenureDays
- if addOns: +10%
- if age > 70: COVID not allowed (we just send message)

Supports:

- multi-destination (Worldwide, Domestic)
- student validation + parent fields
- family vs individual
- add-ons

**Body(example):**

```
{  
  "packageType": "Worldwide",  
  "coverageType": "individual",  
  "startDate": "2025-01-10",  
  "endDate": "2025-01-25",  
  "dob": "1995-05-10",  
  "addOns": ["hijacking", "luggageDelay"]  
}
```

## Submit Proposal

POST /api/travel/submit-proposal

Creates:

- row in travel\_proposals
- rows in travel\_destinations\_selected

### Body(example):

```
{  
  "tripDetails": {  
    "packageType": "Student Travel",  
    "coverageType": "individual",  
    "productPlan": "Student-Plan-1",  
    "startDate": "2025-02-01",  
    "endDate": "2025-02-21",  
    "destinationIds": [1, 2],  
    "addOns": ["hijacking", "luggageDelay"]  
  },  
  "applicantInfo": {
```

```
"firstName": "Ali",
"lastName": "Khan",
"address": "Gulshan-e-Iqbal Block 5",
"cityId": 1,
"cnic": "42101-1234567-1",
"passportNumber": "AB1234567",
"mobile": "03001234567",
"email": "ali@example.com",
"dob": "2000-01-10",
"universityName": "XYZ University"
},
"beneficiary": {
"beneficiaryName": "Danish Khan",
"beneficiaryAddress": "North Nazimabad",
"beneficiaryCnic": "42101-7654321-1",
"beneficiaryCnicIssueDate": "2015-05-01",
"beneficiaryRelation": "Brother"
},
"parentInfo": {
"parentName": "Muhammad Khan",
"parentAddress": "North Nazimabad",
"parentCnic": "42101-1111111-1",
"parentCnicIssueDate": "2010-01-01",
"parentRelation": "Father"
}
}
```

---

## **7.4 My Policies / My Claims (Mock Core)**

### **List Policies**

GET /api/my/policies

Returns from policies\_cache table.

### **List Claims**

GET /api/my/claims

Returns from claims\_cache table.

---

## **7.5 Admin APIs (Backoffice)**

All admin routes require JWT + admin role.

### **Dashboard Summary**

GET /api/admin/dashboard/summary

Returns counts for cards:

- total motor proposals
- total travel proposals
- total payments
- breakdown by status

### **Motor Proposals List**

GET /api/admin/motor-proposals?page=1&limit=20&status=submitted&search=ali

### **Features**

- pagination
- status filter
- search
- default sort newest first (created\_at DESC)

### **Travel Proposals List**

GET /api/admin/travel-proposals?page=1&limit=20&status=paid&search=passport

### **Payments List**

GET /api/admin/payments?page=1&limit=20&status=SUCCESS&search=ORD-TRV&fromDate=2025-01-01&toDate=2025-12-31

---

## **7.6 Notifications / FAQs / Support**

### **Notifications**

- GET /api/notifications
- POST /api/notifications/mark-read

### **FAQs**

- GET /api/faqs

### **Support**

- POST /api/support
  - GET /api/admin/support (admin listing)
- 

## **8. Validation Rules**

Validation is enforced before database inserts. Examples:

### **Motor**

- cnic, cnic\_expiry, dob required
- engine\_number + chassis\_number required
- make\_id, submake\_id, city\_id must exist (FK safe)
- product\_type must be private or commercial

### **Travel**

- start\_date < end\_date
- student rules:
  - if is\_student = 1 → parent fields required

- at least 1 destination must exist in travel\_destinations\_selected
- 

## 9. Error Handling

Common errors:

- 401 Unauthorized → missing/invalid JWT
  - 403 Forbidden → not admin
  - 422 Validation Error → bad input
  - 500 Server Error → unhandled errors (logged)
- 

## 10. Testing

### 10.1 Postman

Use the provided Postman collection JSON to test endpoints.

Flow:

1. Register/Login → copy JWT token
2. Submit Motor/Travel proposal
3. Initiate payment
4. Use admin APIs for verification

### 10.2 Browser Testing (Admin Panel)

Admin panel hits:

- /api/admin/dashboard/summary
  - /api/admin/motor-proposals
  - /api/admin/travel-proposals
  - /api/admin/payments
- 

## 11. Deployment Notes

- Use PM2 or Docker for production runtime

- Set NODE\_ENV=production
  - Store secrets in server env variables
  - Enable HTTPS in production
  - Use MySQL managed service or dedicated VM
- 

## 12. Common Issues

### Foreign Key constraint fails

Means you inserted a record with a referenced id that doesn't exist yet.

Example: motor\_proposals.city\_id must exist in cities.

Fix:

- Seed lookup tables first
  - Ensure correct insert order (parent → child)
- 

## 13. Future Improvements

- Add audit logs for admin actions
  - Add proper file uploads storage (S3/local)
  - Add rate limiting on auth endpoints
  - Add swagger/openapi documentation
- 

### Next (so we make this PERFECT for your repo)

If you paste your **actual route list** (your /src/modules/\*\* files or at least endpoint names), I'll tailor this doc to match your exact API paths + request/response bodies.

Also if you want: I can generate a proper **OpenAPI (Swagger) YAML** for all endpoints (super useful for clients + QA).