SERVISTA – TEST PLAN REPORT

1. Test Coverage Summary

Module	Test Type	Description
User Authentication	Unit + Integration	Signup, login, JWT verification, role-based session handling
Service Catalog	Integration	List services, filter by category, search, view details
Booking Management	Unit + Integration	Create booking, assign provider, update booking status, cancel booking
Provider Dashboard	Integration	Accept jobs, update status, view assigned tasks
Payment Gateway	Integration	Razorpay test mode payments, order creation, signature verification
Admin Dashboard	System + Integration	KPIs, analytics, users list, bookings & payments monitoring
Role-Based Access	Unit	Access control for Customer, Provider, Admin
Database (PostgreSQL)	Unit	CRUD operations, schema validation, constraints
API Endpoints	Integration	REST API response validation (status code, format, error handling)
UI Testing (Optional)	Functional/UI	Form validation, responsiveness, navigation, error boundaries

2. Test Cases

USER AUTHENTICATION TESTS

Test ID	Test Case	Steps	Expected Result
UA-01	Register new customer	POST `/auth/register` with valid name, email, password	`201 Created` — "User registered successfully"
UA-02	Duplicate registration	Send same email again	`400 Bad Request` — "User already exists"
UA-03	Login valid user	POST `/auth/login` with correct credentials	Returns JWT token + user role
UA-04	Login invalid password	Wrong password	`401 Unauthorized` — "Invalid credentials"
UA-05	Access protected route	Call `/user/profile` with valid token	Returns user profile
UA-06	No token provided	Call protected route without token	`401 Unauthorized`

SERVICE CATALOG TESTS

Test ID	Test Case	Steps	Expected Result
SC-01	Get all services	GET `/services/all`	Returns list of services
SC-02	Search service	GET `/services/search?query=cleaning`	Returns matching services
SC-03	Invalid service ID	GET `/services/123abc`	`404 Not Found`
SC-04	Filter by category	GET `/services/category/plumbing`	Returns category-specific list

BOOKING MANAGEMENT TESTS

Test ID	Test Case	Steps	Expected Result
BK-01	Create booking	POST `/booking/create` with valid data	`201` Booking created
BK-02	Overlapping booking	POST same date/time	`409` — “Slot already booked”
BK-03	Cancel booking	DELETE `/booking/cancel/:id`	`200` — “Booking cancelled”
BK-04	Fetch booking history	GET `/booking/my-bookings`	Returns list of customer bookings
BK-05	Unauthorized booking	No token in header	`401 Unauthorized`
BK-06	Provider assignment	Auto assignment triggered	Provider receives booking in dashboard

PROVIDER DASHBOARD TESTS

Test ID	Test Case	Steps	Expected Result
PD-01	View assigned bookings	GET `/provider/bookings`	Returns pending & ongoing jobs
PD-02	Accept booking	POST `/provider/accept/:id`	`200` — Status updated
PD-03	Update job status	POST `/provider/update-status/:id`	Status changes: Pending → In-progress → Completed
PD-04	Unauthorized access	Customer accessing provider routes	`403 Forbidden`

PAYMENT GATEWAY TESTS (RAZORPAY)

Test ID	Test Case	Steps	Expected Result
PG-01	Create Razorpay order	POST `/payment/order`	Returns order ID
PG-02	Complete payment	Complete Razorpay test mode payment	Returns `Payment successful`
PG-03	Invalid signature	Tamper signature	`400` — “Payment verification failed”
PG-04	Fetch payment status	GET `/payment/status/:id`	Returns payment details

ADMIN DASHBOARD TESTS

Test ID	Test Case	Steps	Expected Result
AD-01	View all users	GET `/admin/users`	Returns list of customers & providers
AD-02	View all bookings	GET `/admin/bookings`	Shows all platform bookings
AD-03	Analytics charts	GET `/admin/stats`	Returns KPIs: revenue, bookings per day
AD-04	Delete provider/user	DELETE `/admin/delete-user/:id`	Record deleted

ROLE-BASED ACCESS TESTS

Test ID	Test Case	Steps	Expected Result
RB-01	Customer accessing admin route	GET `/admin/users`	`403 Access denied`
RB-02	Provider accessing customer route	GET `/user/bookings`	`403 Access denied`
RB-03	Admin accessing all routes	Admin token	`200 Success`

DATABASE TESTS (PostgreSQL)

Test ID	Test Case	Steps	Expected Result
DB-01	Insert booking record	Insert valid booking row	Row inserted successfully
DB-02	Missing mandatory fields	Insert invalid row	Constraint/validation error
DB-03	Delete user record	Remove by ID	Cascading deletes handled (if defined)
DB-04	Unique constraints	Duplicate email	Error: duplicate key value

3. Testing Approach

✔ Manual Testing

- Used Postman to test all API endpoints
- Tested frontend UI for user flows
- Razorpay test mode to simulate payments

✔ Tools Used

- **Postman** – API Testing
- **pgAdmin** – PostgreSQL monitoring
- **Browser DevTools** – UI debugging

4. Validation Checklist

- All REST APIs tested (status codes, responses)
- All CRUD operations functioning
- Role-based access verified
- Payment flow completed successfully
- UI forms validated with both valid & invalid inputs
- SQL errors gracefully handled
- Booking business rules validated
- Provider workflow validated