

```
File Edit View
2026-02-14 01:32:00 INFO [hooks.ScenarioHook] Launching Chrome Browser
2026-02-14 01:32:05 INFO [stepdefinition.BookingTest] Opening Airbnb website
2026-02-14 01:32:10 INFO [stepdefinition.BookingTest] Searching Destinations
2026-02-14 01:32:20 INFO [stepdefinition.BookingTest] Typed destination: Goa
2026-02-14 01:32:27 INFO [stepdefinition.BookingTest] Selected Checkin dates
2026-02-14 01:32:33 INFO [stepdefinition.BookingTest] Selected Checkout dates
2026-02-14 01:32:40 INFO [stepdefinition.BookingTest] Clicked on Searched Button
2026-02-14 01:32:44 INFO [hooks.ScenarioHook] Scenario PASSED: Validate booking price filter works
2026-02-14 01:32:44 INFO [hooks.ScenarioHook] Closing browser...
```

src/test/resources/features/booking.feature:3

Passed Validate booking price filter works

Overview History Retries

Severity: normal

Duration: 45s 044ms

Execution

▼ Set up

- ✓ **stepdefinition.BookingTest.initPages()** 624ms
- ✓ **hooks.ScenarioHook.setup()** 5s 304ms

▼ Test body

- ✓ Given user is on the booking site home page 17s 958ms
- ✓ When user searches for "Goa" 2ms
- ✓ And user selects check-in "2026-03-10" and check-out "2026-03-12" 19s 891ms
- ✓ And user applies price filter "1000" to "3000" 1ms
- ✓ Then results should display only items within "1000" to "3000" 1ms

▼ Tear down

- ✓ **hooks.ScenarioHook.tearDown(io.cucumber.java.Scenario)** 1s 251ms

✓ Success Screenshot

547 Kib

Allure

Suites

order name duration status

Status: 0 0 1 0 0 Marks: ✅ ✅ ✅ ✅ ✅

Booking functionality

#1 Validate booking price filter works 45s 044ms

Test body

- Given user is on the booking site home page 17s 958ms
- When user searches for "Goa" 2ms
- And user selects check-in "2026-03-10" and check-out "2026-03-12" 19s 891ms
- And user applies price filter "1000" to "3000" 1ms
- Then results should display only items within "1000" to "3000" 1ms

Tear down

- hooks.ScenarioHook.tearDown(io.cucumber.java.Scenario) 1s 251ms

Success Screenshot 547 KiB

Over 1,000 homes

Flat in Vagator Flat in Candolim

Now you'll see one price for your trip, all fees included.

Got it

vd

Slide 1 — Project Title

Booking Website Automation Framework (Airbnb/Booking)

Built using **Selenium, TestNG, Cucumber BDD, POM, Allure Reports, Log4j2** with **Java + Maven**.

★ Slide 2 — Project Objective

- Automate the **search, date selection, price filter, and results validation** flow of a booking website.
- Build a **complete automation framework** with professional tools:
 - Cucumber BDD
 - POM (Page Object Model)
 - TestNG
 - Allure Reporting
 - Log4j2 Logging
- Provide **readable tests, reusable components, screenshots, and logs**.

★ Slide 3 — Tools & Technologies Used

Languages & Build

- Java
- Maven

Automation & Testing Tools

- Selenium WebDriver
- Cucumber BDD (Gherkin)
- TestNG (runner + assertions + suite control)
- Page Object Model

Reporting & Monitoring

- Allure Report
- Log4j2 Logs (automation.log & error.log)

Others

- ChromeDriver
- Thread waits, XPaths, properties file

★ Slide 4 — Framework Architecture

```
Final_Project_01
├── src/main/java
│   ├── base           → Base.java (WebDriver reference)
│   └── pageObject     → HomePage.java (POM methods)
├── src/test/java
│   ├── cucumberoption → TestRunner.java (Cucumber + TestNG)
│   ├── hooks          → TestHooks.java (Before/After, screenshots, logs)
│   └── stepdefinition → BookingSteps.java (step definitions)
└── src/test/resources
    ├── features        → booking_search_and_filter.feature
    ├── log4j2.xml       → Log configuration
    └── data.properties  → URL, test data
└── logs              → automation.log, error.log
```

★ Slide 5 — Cucumber BDD (Feature)

Gherkin scenario covered

- Open booking page
- Enter destination
- Select check-in & check-out dates
- Apply price filter
- Validate results between min–max price

Advantages

- Readable test cases
- Business-friendly language
- Easy collaboration

★ Slide 6 — Page Object Model (POM)

HomePage.java

- Contains all locators (XPaths)
- Contains reusable methods:
 - openCalendar()
 - clickByXpath()
 - selectDates()
 - enter destination
 - apply filters

Benefits

- Reusable
- Reduces duplicate code
- Simple maintenance
- Better readability

★ Slide 7 — Step Definitions

BookingSteps.java

Handles:

- Search box actions
- Date selection
- Price filter
- Validation

- Calls POM methods → maintains clean separation

Hooks

- TestHooks manages:
 - WebDriver startup
 - WebDriver quit
 - Success & failure screenshots
 - Attaching logs to Allure
 - Logging scenario status

★ Slide 8 — Allure Reporting

What you added

- allure-cucumber7-jvm
- Attached screenshot on pass & fail
- Attached automation logs
- Generated report using: `allure serve target/allure-results`

Allure Features Used

- Detailed steps
- Screenshots
- Logs
- Timeline
- Scenario history
- Attachments
- Pretty UI with graphs

★ Slide 9 — Logging (Log4j2)

What you implemented

- automation.log (INFO, WARN, ERROR)
- error.log (only errors)
- Logs understand:
 - Browser launch
 - URL navigation
 - Clicking elements
 - Date selection
 - Failures

Benefits

- Debug failures easily
- Tracks execution flow
- Helps during development + production
- Integrated with Allure

★ Slide 10 — Key Automation Achievements

- ✓ Built complete automation framework from scratch
- ✓ Successfully automated booking workflow
- ✓ Implemented reusable Page Object Model
- ✓ Added Cucumber BDD readability
- ✓ Integrated logs (Log4j2)
- ✓ Integrated Allure Reports with screenshots
- ✓ Clean folder structure
- ✓ Professional test execution workflow
- ✓ Modern best-practices implementation

★ Slide 11 — What I Learned

- XPath building
- POM architecture
- Cucumber step glue
- TestNG execution
- Creating Allure reports
- Generating logs
- Debugging using logs + screenshots
- Maven dependency handling
- Setting up testing frameworks in Eclipse

★ Slide 12 — Final Outcome

A fully functional, professional-level automation framework with:

- BDD scenarios
- POM structure
- Logging system
- Screenshots (pass + fail)
- Allure reports
- Clean & scalable architecture
- Real-world booking site automation