🏨Hotel Reservation System – Full Setup & Error Resolution Guide

---

Project Directory Structure

```
HotelReservationSystem/
├── frontend/
│   ├── index.html
│   └── pages/
│       ├── reserve/
│       │   ├── reserve.html
│       │   ├── reserve.css
│       │   ├── BookingServlet.java
│       │   └── confirm.jsp
│       └── viewbooking/
│           └── viewbooking.html
│
├── WEB-INF/
│   ├── web.xml
│   └── classes/   (compiled .class files go here)
```

Tech Stack Summary

Layer	Technology	Purpose
🕋Frontend	HTML, CSS	User Interface design
Backend	Java (Servlets), JSP	Business Logic, Routing
Storage	File I/O or SQLite (optional)	Booking persistence (extendable)

Compilation Guide – Fixing `` Errors

❌Error:

```
error: package jakarta.servlet does not exist
error: cannot find symbol: class HttpServlet
```

✅Cause:

You included only 1 `.jar` file. Servlets require **multiple JARs**.

✅Solution:

1. Open Command Prompt (CMD)
2. Navigate to your project folder:

```
cd E:\PROJECTS\HotelReservationSystem
```

3. Compile using all required JARs:

```
javac -cp "C:\Needy\glassfish-7.0.25\glassfish7\glassfish\modules\*" -d
WEB-INF\classes frontend\pages\reserve\BookingServlet.java
```

This compiles the servlet successfully.

If `` doesn't work, use full JAR paths:

```
javac -cp "C:
\Needy\glassfish-7.0.25\glassfish7\glassfish\modules\jakarta.servlet-api.jar;C:
\Needy\glassfish-7.0.25\glassfish7\glassfish\modules\jakarta.annotation-
api.jar;C:\Needy\glassfish-7.0.25\glassfish7\glassfish\modules\jakarta.inject-
api.jar" -d WEB-INF\classes frontend\pages\reserve\BookingServlet.java
```

🚀GlassFish Server Setup

✔️Set Java Path (if needed):

```
set JAVA_HOME=E:\PROJECTS\JDK
set PATH=%JAVA_HOME%\bin;%PATH%
```

▶️Start Server:

```
cd C:\Needy\glassfish-7.0.25\glassfish7\bin
asadmin start-domain
```

🔄Restart Server (if port already in use):

```
asadmin stop-domain
asadmin start-domain
```

Deploying the Project

Option 1: Deploy via Admin Console

1. Open browser → `http://localhost:4848`
2. Navigate: **Applications > Deploy**
3. Choose either project folder or `.war` file

Option 2: Create WAR File (optional)

```
jar -cvf HotelReservationSystem.war *
```

Then deploy `.war` via Admin Console.

Important URLs to Access Pages

Page	URL
🏠Home	`http://localhost:8080/HotelReservationSystem/frontend/index.html`
Booking Form	`http://localhost:8080/HotelReservationSystem/frontend/pages/reserve/reserve.html`
View Bookings	`http://localhost:8080/HotelReservationSystem/frontend/pages/viewbooking/viewbooking.html`
🚃Servlet Route	`/HotelReservationSystem/booking` (BookingServlet.java)

🚃Checklist: Every Time You Restart Your PC

1. Set Java environment:

```
set JAVA_HOME=E:\PROJECTS\JDK
set PATH=%JAVA_HOME%\bin;%PATH%
```

1. Start GlassFish server:

```
cd C:\Needy\glassfish-7.0.25\glassfish7\bin
asadmin start-domain
```

1. Recompile servlets if updated:

```
javac -cp "C:\Needy\glassfish-7.0.25\glassfish7\glassfish\modules\*" -d WEB-INF\classes frontend\pages\reserve\BookingServlet.java
```

1. Visit site:

```
http://localhost:8080/HotelReservationSystem/frontend/index.html
```

Tips

- Place `.class` files inside `WEB-INF/classes` after compilation.
- Use `web.xml` for servlet mapping.
- Avoid using old servlet JARs like `javax.servlet` in Jakarta EE 10+.
- For quick testing, always restart domain after major deployment.

You're all set to build, deploy, and scale your Java Servlet-based Hotel Reservation System! Need a PDF or DOC version of this guide? Just ask!

Pro Tip: For JSP and Servlet debugging, use browser DevTools and `System.out.println()` in servlet code for simple logs.