Smart Health Monitoring System - 9 Commands with Theory & Code

1. Creating the New Project with JDK & IDE Setup

Theory: Setting up your development environment with Java (JDK 11+) and IntelliJ/Eclipse ensures that your Smart Health Monitoring System runs efficiently.

Coding:

- Install JDK 11+.
- In IntelliJ: File > New > Project > Java > Choose JDK > Finish
- Optionally include JavaFX libraries if GUI is used.
- Add MySQL JDBC driver for database integration.

2. Define the Project Structure

Theory: A well-structured project increases readability and maintainability.

Suggested Structure:

- model: User, Patient, Vitals
- dao: UserDAO, VitalsDAO
- controller: LoginController, VitalsController
- util: DBConnection, ValidationUtils

3. Design the Database Schema

Theory: A schema is essential for storing users, vitals, and other system information.

Example Tables:

```
CREATE TABLE users (...);
CREATE TABLE vitals (...);
```

4. Create MySQL Tables

Theory: These tables will hold patient data and system users.

Execution:

Smart Health Monitoring System - 9 Commands with Theory & Code

Use MySQL Workbench or CLI to run the schema SQL queries.

5. Implement JDBC for Database Connectivity

Theory: JDBC connects your Java app to MySQL for data operations.

Coding:

```
public class DBConnection { ... }
```

6. Create Model and DAO Classes

Theory: Model classes represent your data; DAOs handle database interaction.

Example:

```
public class Vitals { ... }
public class VitalsDAO { ... }
```

7. Aesthetics and Visual Appeal of the UI

Theory: A clean UI ensures a better experience, especially for patients and medical staff.

Tips:

- Use blue and green tones
- Group similar elements
- Use icons
- JavaFX CSS styling

8. Component Placement and Alignment

Theory: Proper layout supports readability and usability.

Example:

```
GridPane grid = new GridPane();
grid.add(new Label("Heart Rate:"), 0, 0); ...
```

Smart Health Monitoring System - 9 Commands with Theory & Code

9. Responsiveness and Accessibility

Theory: Your GUI should adapt to various screen sizes and be usable for everyone.

Tips:

- Use VBox, HBox, GridPane

- Include keyboard navigation

- Use Label.setLabelFor() for screen readers