The AdminController code was implemented by injecting key dependencies—AppDbContext for data access and IHubContext<ClaimsHub> for SignalR messaging—using constructor injection, following standard ASP.NET Core design principles that promote testability and separation of concerns.

The controller exposes two main actions:

Index(): Retrieves claims from the database ordered by submission date and returns them to the view.

UpdateStatus(int id, string action): Updates the status of a claim based on the provided action string, using an enum ClaimStatus for type safety and clear representation of possible statuses (Approved, Rejected, Pending). This method then saves changes and notifies clients via SignalR.

Improvements made include:

Enum Usage: The status updates use the ClaimStatus enum directly instead of string literals, improving code clarity and preventing invalid status values.

Correct Enum Assignment: Fixed the incorrect call ClaimStatus.ToString(Approved) to directly assign enum members (ClaimStatus.Approved), avoiding implicit conversion errors.

ToString() Usage: When sending messages, the enum value is converted to string with .ToString() for display or transmission purposes.

Single Declaration of Dependencies: Private members like \_db and \_hub are declared only once, eliminating ambiguity errors (CS0229) that came from duplicate declarations.

Dependency Injection: Constructor injection is used to provide dependencies, enabling easier testing and adherence to SOLID principles.

Robust Null Handling: The update method correctly returns NotFound() if the claim is missing, improving error handling.

Overall, the implementations were improved by adhering to best practices in enum handling, dependency injection, error handling, and clean coding standards. This makes the controller robust, maintainable, and easier to test.

Enums were leveraged to represent fixed status states with clarity, and SignalR integration ensures real-time notifications on state changes. The code transition from incorrect to proper enum assignments addresses a common source of runtime bugs and compiler errors, elevating code quality significantly.