

**FitCheck App Product Requirements Document**

[**1. Introduction 3**](#_g5cy2hn6bmik)

[1.1. Purpose 3](#_t6dm10h9zo5q)

[1.2. Scope 3](#_mh02x6mfxzle)

[1.3. Definitions, Acronyms, and Abbreviations 3](#_93ahdwwpen0k)

[**2. Functional Requirements 3**](#_z4iy9m8dw2y9)

[2.1. User Features 3](#_kitcifz7ulaf)

[2.1.1. Account Creation & Profile Management 3](#_fozyl6nxxohh)

[2.1.2. Size Recommendation 5](#_esxzerr255cd)

[2.1.3. Feedback 5](#_heqlzy5kiw7o)

[2.2. Admin Features 6](#_otqt4m3fn72l)

[2.2.1. User Management 6](#_3y50mrig1ztv)

[2.2.2. Clothing Database Management 6](#_ys3i5v9mvxmh)

[2.2.3. Analytics & Insights 7](#_fznv7ijqw0t7)

[2.3. Machine Learning Algorithm 7](#_ra5lzftedox3)

[2.4. Pages 8](#_v2gdghvbnvb6)

[2.4.1. User Pages 8](#_h3bycof3y9j)

[2.4.2. Admin Pages 9](#_p3n6jyqkut2)

[2.4.3. Additional Pages 9](#_uvgu4lsx2dc)

[**3. Technical Requirements 9**](#_tur34qlk0l7s)

[3.1. Front-End 9](#_auhb5z7sz4oz)

[3.2. Back-End 10](#_m0tbgroft8n3)

[3.3. Machine Learning 10](#_5fiam2a6db2l)

[3.4. Cloud Services 10](#_y1ldygnc78wj)

[3.5. User Details Storage 10](#_2busyd4ffcfa)

[3.6. Clothing Database 11](#_oh9o84jrk34h)

[3.7. Integration Strategy 12](#_r62zbam3ph09)

[3.8. Model Processing Flow 12](#_hdneugr92cu8)

# 

# 

# 

# 

# Introduction

## Purpose

The purpose of the FitCheck application is to revolutionize the online apparel shopping experience by providing users with accurate clothing size recommendations based on their unique body measurements extracted from uploaded images and videos. By leveraging advanced machine learning algorithms and seamless integration with cloud services, FitCheck aims to reduce the rate of apparel returns due to size discrepancies, thereby enhancing user satisfaction and improving overall shopping efficiency.

## Scope

The scope of FitCheck includes user-centric features for account management, personalized size recommendations, and administrative functionalities for managing user accounts and the clothing database. Integration of machine learning algorithms facilitates precise measurement extraction and size prediction. Technical requirements involve Next.js and NestJS for frontend and backend development respectively, integration with AWS services for data storage and processing, and deployment of TensorFlow models for image and video analysis.

## Definitions, Acronyms, and Abbreviations

* **AI:** Artificial Intelligence
* **PRD:** Product Requirements Document
* **UI:** User Interface
* **ML:** Machine Learning
* **API:** Application Programming Interface
* **UX:** User Experience

# Functional Requirements

## User Features

### Account Creation & Profile Management

* **Description:** Users can create an account input, upload body-size related information and manage their profiles.
* **Functionalities:**
  + **User Registration Form:**
    - Full Name
    - Email Address
    - Password
  + **Profile Management:**
    - **Text-Based Data Collection:**
      * Input fields for height (cm), weight (kg), and body type (with visual examples).
      * Data validation and secure storage in the database.
      * Options to update and manage personal information, including name and password changes.
  + **Image-Based Data Collection:**
    - **Image Upload:**
      * Functionality for users to upload front and side pictures.
      * Real-time validation for image quality and inclusion of reference objects (A4 paper or credit card).
      * Image processing to calculate body proportions and measurements.
      * Secure storage of images and extracted data in the database, with the option to delete images after extraction.
  + **Video-Based Data Collection:**
    - **Video Upload:**
      * Upload functionality for 360-degree videos (supported formats: MP4, AVI).
      * Input fields for height and weight.
      * Video processing to extract body proportion data.
      * Secure storage of videos and extracted data in the database, with the option to delete videos after extraction.
  + **Error Handling:**
    - Notifications for invalid uploads.
    - Guidance for users to correct issues and ensure quality uploads.
  + **Update Profile Information**
    - **Edit Personal Details:**
      * Change Full Name and Email Address.
      * Update password for account security.
    - **Update Body Measurements:**
      * Modify height and weight details.
      * Re-upload front and side images for updated measurements.
      * Re-upload 360-degree videos for accurate body proportion analysis.
    - **Real-Time Validation:**
      * Ensure updated image and video data meet quality standards for accurate processing.

### Size Recommendation

* **Description:** Provides users with personalized size recommendations for clothing.
  + **Functionalities:**
  + **Size Comparison:**
    - Uses extracted body measurements from user inputs (images, videos, height, weight).
    - Compares user's dimensions against clothing item measurements stored in the database.
  + **Database Query:**
    - Accesses a test user database to find individuals with similar body dimensions.
    - Analyzes historical fit data to identify successful matches and recommendations.
  + **Recommendation Calculation:**
    - Calculates the best-fitting size using a combination of:
      * User-specific body data.
      * Clothing item dimensions.
      * Historical data from similar users.
  + **Real-Time Updates:**
    - Provides size recommendations while users browse clothing on partner web stores.
    - Dynamically adjusts suggestions based on updated user profile data or newly available clothing data.

### Feedback

* **Description:** Users can provide feedback on clothing fit to improve the recommendation algorithm.
* **Functionalities:**
  + **User Feedback Submission:**
    - Users provide feedback on how well the recommended sizes fit.
    - Simple and intuitive interface for feedback collection.
  + **Integration with ML Model:**
    - User feedback is utilized to enhance the accuracy of the size recommendation algorithm.
    - Continuously updates the model based on new feedback and data to improve recommendations.

## Admin Features

### User Management

* **Description:** Admins can manage user accounts and profiles.
* **Functionalities:**
  + **Admin Dashboard:**
    - View and manage user accounts efficiently, to support with user inquiries.

### Clothing Database Management

* **Description:** Admins manage the database of clothing items.
* **Functionalities:**
  + **CRUD Operations:**
    - **Add, Update, and Remove Clothing Items:**
      * Seamlessly add new clothing items with detailed specifications.
      * Update existing items with new measurements or details.
      * Remove outdated or discontinued items from the database.
  + **Data Validation:**
    - **Accuracy of Clothing Measurements:**
      * Validate clothing measurements and details before adding to the database.
      * Ensure consistent categorization by gender, type, and size.
      * Use automated checks to confirm data integrity and correctness.
  + **Inventory Categorization:**
    - Organize items by gender, type, and available sizes.
    - Maintain a structured database for easy retrieval and updates.
  + **Measurements:**
    - Provide measurements in centimeters for each size.
    - List available sizes (e.g., S, M, L, 32/32, 29/30).
  + **Integration:**
    - Ensure efficient data retrieval for user size recommendations.
    - Support seamless integration with the size recommendation engine.

### Analytics & Insights

* **Description:** Admins can access detailed analytics on user engagement and platform performance.
* **Functionalities:**
  + **User Metrics:**
    - **Active Users:** Track the number of active users over various time periods (daily, weekly, monthly).
    - **Feedback:** Analyze user feedback on clothing fit and platform experience.
  + **Clothing Item Metrics:**
    - **Database Activity:** Monitor additions, updates, and removals of clothing items.
    - **Popularity Trends:** Identify popular items based on user interactions and recommendations.
  + **Recommendation Accuracy:**
    - **Effectiveness Analysis:**
      * Use user feedback on fit to assess the accuracy of size recommendations.
      * Analyze return rates and reasons for returns to identify potential sizing issues.
      * Evaluate historical data comparing user dimensions and recommended sizes with actual fits.

## Machine Learning Algorithm

* **Description:** The system utilizes machine learning to recommend clothing sizes based on user data.
* **Functionalities:**
  + **Data Preprocessing and Feature Extraction:**
    - Extract relevant features from user profiles, including text inputs, images, and videos.
    - Process user-provided images and videos to determine precise body measurements.
    - Analyze clothing item data for measurements and available sizes.
  + **Model Training and Recommendations:**
    - Train the model using historical user data and feedback on clothing fit.
    - Use test user data to refine the model’s accuracy.
    - Generate size recommendations by comparing extracted user measurements with the clothing database.
  + **Continuous Improvement:**
    - Continuously update the model with new user feedback and data.
    - Improve recommendation accuracy based on user feedback regarding clothing fit.
  + **Integration with Size Recommendation Engine:**
    - Provide real-time size suggestions during user browsing.
    - Ensure efficient matching of user dimensions with clothing item measurements for optimal fit.

## Pages

### User Pages

* **Homepage**
  + Overview of the FitCheck application and its features.
  + Navigation to other sections.
* **Account Management**
  + **Login**
    - Form for existing users to log into their accounts.
  + **Registration**
    - Form for new users to create an account.
  + **Profile Dashboard**
    - Overview of user's profile information and options to manage account settings.
* **Measurement Input**
  + **Text-Based Input**
    - Form for users to input their height, weight, and body type.
  + **Image-Based Input**
    - Upload interface for front and side images for body measurements.
  + **Video-Based Input**
    - Upload interface for 360-degree videos to extract body measurements.
* **Size Recommendation**
  + **Recommendation Page**
    - Displays personalized clothing size recommendations based on user inputs.
  + **Comparison Tool**
    - Allows users to compare their measurements with clothing item dimensions.
* **Feedback**
  + **Feedback Submission**
    - Form for users to provide feedback on clothing fit and overall experience.

### Admin Pages

* **Admin Dashboard**
  + Overview of platform metrics and administrative tools.
* **User Management**
  + **User List**
    - View and manage user accounts.
  + **User Details**
    - Detailed view of user profiles and activity.
* **Clothing Management**
  + **Inventory Management**
    - CRUD operations for adding, updating, and removing clothing items.
  + **Measurement Validation**
    - Interface for validating and updating clothing measurements.
* **Analytics & Insights**
  + **User Engagement**
    - Metrics on user activity, feedback submissions, and engagement.
  + **Clothing Item Analytics**
    - Performance metrics for clothing items, popularity trends, and fit analysis.

### Additional Pages

* **About Us**
  + Information about the FitCheck application and its team.
* **Contact**
  + Contact form or information for user inquiries.
* **Privacy Policy**
  + Detailed policy on user data protection and usage.
* **Terms of Service**
  + Legal terms governing the use of the FitCheck application.

# Technical Requirements

## Front-End

### Framework: Next.js

* **Reasons:**
  + **Fast Rendering:** Supports both server-side rendering (SSR) and incremental static generation (ISG), improving performance and SEO.
  + **Easy Integration:** Seamlessly integrates with AWS services for backend communication and data handling.
  + **Built-in Routing:** Simplifies routing and navigation with a file-based system.
  + **TypeScript Support:** Built-in TypeScript support for type safety and improved developer experience.

### Alternative: React

* **Reasons:**
  + **Component-Based Architecture:** Promotes reusability and modularity.
  + **Large Ecosystem:** Extensive library support for state management, UI components, and other functionalities.
  + **Flexible:** Can be integrated with various rendering libraries to achieve SSR (e.g., Next.js, Gatsby).
  + **Cons**
    - Requires additional configuration for SSR/ISG, which can add complexity, such as performance tuning and debugging.
    - React is not inherently type-safe, which can lead to runtime errors, reduced maintainability, and slower development due to the lack of compile-time checks.

## Back-End

* + **Framework:** NestJS
    - **Reasons:**
      * Scalable architecture with TypeScript support.
      * Integration capabilities with AWS services (Lambda, S3, RDS) for server-side logic and data management.
      * REST API development for communication between front-end and AWS backend services.
  + **Alternative: Express.js**
    - **Reasons:**
      * **Lightweight and Flexible:** Minimalist framework, easy to set up and customize.
      * **Large Community:** Extensive resources and middleware available.
      * **Quick Prototyping:** Ideal for small to medium-sized projects.
      * **Cons:**
        + Less opinionated structure, which can lead to fragmented code in larger projects.
        + Not type safe, which may result in runtime errors.
        + Middleware-heavy, which can complicate debugging.
        + Less out-of-the-box functionality compared to more structured frameworks.

## Machine Learning

* + **Language:** Python
  + **Framework:** TensorFlow
    - **Reasons:**
      * **GPU Acceleration:** Optimized for high-performance computing.
      * **Production Ready:** Strong deployment capabilities on various platforms.
      * **Extensive Library Support:** Pre-trained models and tools for computer vision, NLP, etc.
  + **Alternative: PyTorch**
    - **Reasons:**
      * **Dynamic Computation Graph:** Easier debugging and flexible model building.
      * **Intuitive API:** More Pythonic and user-friendly.
      * **Strong Research Community:** Preferred in academia and for prototyping.
      * **Cons:**
        + **Deployment Complexity:** Historically more challenging to deploy models into production compared to TensorFlow.
        + **Scalability Challenges:** Scaling for large production workloads may require more optimization and effort.
        + **Limited Built-in Tools:** Fewer pre-built models and tools available compared to some competitors.
        + **Community Adoption:** While growing, historically had less industry adoption compared to TensorFlow, impacting support and ecosystem.

## Cloud Service Infrastructure

* + **Platform:** AWS (Amazon Web Services)
    - **Services:**
      * **S3 (Simple Storage Service):** Storage for user-uploaded images and videos.
      * **Lambda:** Serverless execution for image and video processing, integrating with S3 for scalable processing.
      * **RDS (Relational Database Service):** PostgreSQL or MySQL database for storing user profiles, measurements, and application data securely.
      * **EC2 (Elastic Compute Cloud):** Hosting backend server and machine learning model, ensuring scalable computing capacity.
      * **API Gateway:** Facilitating communication between front-end and backend services, managing REST APIs for data retrieval and processing.
      * **CloudFront:** Content delivery network (CDN) for fast content delivery globally, enhancing user experience.
      * **Elasticsearch:** Supporting efficient search functionality for clothing items based on user preferences and characteristics.
      * **CloudWatch:** Monitoring and logging for the entire AWS infrastructure, ensuring operational health and performance.
  + **Alternative: Google Cloud Platform (GCP)**
    - **Services:**
      * **Google Cloud Storage:** Similar to S3, providing object storage with high availability and performance.
      * **Cloud Functions:** Equivalent to Lambda, supporting serverless event-driven computing.
      * **Cloud SQL:** Managed databases with support for PostgreSQL/MySQL, similar to RDS.
      * **Compute Engine:** Virtual machines similar to EC2, offering flexibility and scalability.
      * **API Gateway:** Manages APIs, similar to AWS’s API Gateway.
      * **Cloud CDN:** Content delivery network for fast and secure web content delivery.
      * **BigQuery:** Powerful analytics data warehouse, great for large-scale data analysis.
      * **Stackdriver:** Monitoring and logging similar to CloudWatch.
      * **Cons:** Less mature ecosystem for general web applications compared to AWS, but excellent for data processing and machine learning. Our development team also has less experience with this platform.

## User Details & Clothing Storage

* + **Database:** RDS (PostgreSQL or MySQL)

### User Details

* + - **Schema:**
      * **Users Table:**
        + User ID (Primary Key)
        + Full Name
        + Email Address (Unique)
        + Password (Hashed)
        + Registration Date
        + Last Login Date
        + Profile Picture (Stored in S3 with reference in RDS)
        + Body Measurements (Height, Weight)
        + Measurements Extracted by ML:

Waist circumference

Hip circumference

Inseam length

Chest circumference (if applicable)

Others as determined by ML model

* + - * + Uploaded Images/Videos (References or direct links stored in RDS)
    - **Functionality:**
      * CRUD operations for managing user accounts, including registration, login, and profile updates.
      * Secure password hashing and storage.
      * Integration with AWS Cognito or custom authentication for user authentication and session management.
      * Storage of user-uploaded images/videos and associated metadata in S3, with references stored in RDS for efficient retrieval.

### Clothing Storage

* + - **Schema:**
      * **Clothing Items:**
        + Item ID
        + Name
        + Description
        + Product Image (Stored in S3 with reference in RDS)
        + Measurements (e.g., chest, waist, length)
        + Available Sizes (e.g., S, M, L, XL)
        + Gender
        + Category (e.g., shirts, pants, dresses)
        + Material and Care Instructions
        + Historical Fit Data (user feedback on fit)
    - **Functionality:**
      * CRUD operations for adding, updating, and removing clothing items.
      * Integration with the size recommendation engine to match user measurements with clothing item dimensions.
      * Data validation to ensure accurate measurements and details before adding to the database.

### 

## Integration Strategy

* **Front-End Integration:**
  + Utilize Next.js for building responsive and SEO-friendly web interfaces, integrating seamlessly with AWS services for dynamic content and data retrieval.
* **Back-End Integration:**
  + Implement NestJS controllers and services for managing AWS service interactions, ensuring efficient data flow and processing between front-end user interactions and backend logic.
* **Machine Learning Model Deployment:**
  + Deploy TensorFlow models on AWS Lambda for processing user-uploaded images and videos, with results stored securely in AWS RDS.

## Model Processing Flow

* **User Uploads:**
  + Users upload images (front and side views) or videos (360-degree view) containing their body.
* **Preprocessing:**
  + Uploaded images/videos undergo preprocessing steps to ensure they meet quality standards and are suitable for analysis.
* **Feature Extraction:**
  + For image-based inputs:
    - The model extracts key features such as body dimensions (e.g., waist, hip, inseam) from the images using computer vision techniques.
  + For video-based inputs:
    - Frames from the video are extracted and processed to derive comprehensive body measurements in a 360-degree view.
* **Model Deployment:**
  + The machine learning model (developed using TensorFlow or PyTorch) is deployed on a cloud-based service, such as AWS Lambda or EC2, depending on computational requirements and scalability needs.
* **Inference:**
  + When a user uploads their data, the backend server sends this data to the deployed machine learning model for inference.
  + The model performs calculations and predictions based on the input data to determine precise body measurements.
* **Result Storage:**
  + The processed data (extracted body measurements) is securely stored in the RDS (Relational Database Service) on AWS, associated with the user’s profile.