# SkillByte - Product Requirements Document (PRD)

## Overview

SkillByte is a mobile application for micro-learning through short-form videos (60-90 seconds). Users can learn practical skills with structured progression and community engagement.

## Target Audience

- Primary: Young professionals (25-35)

- Secondary: Students (16-25)

- Tertiary: Lifelong learners (35+)

## Core Features (MVP - Week 1)

### Authentication

- Email/password signup and login

- Google authentication

- Profile creation and management

- Session handling

- Password recovery

- User profile data storage

### Video Feed

- Vertical scroll feed

- Auto-play on focus

- Progress indicator

- Loading states

- Error handling

- Play/pause functionality

- Progress bar

- Sound control

- Full-screen mode

- Quality selection

### Content Management

- Video requirements:

\* Duration: 60-90 seconds

\* Max file size: 50MB

\* Formats: MP4, MOV

\* Resolution: up to 1080p

- Metadata:

\* Title

\* Description

\* Category/tags

\* Skill level

\* Prerequisites

### User Interaction

- Like/save videos

- Share functionality

- Progress tracking

- Completion marking

- Basic comments

- Progress metrics:

\* Videos watched

\* Skills started

\* Completion rates

\* Time invested

### Skill Categories

- Main categories:

\* Professional Skills

\* Communication

\* Problem Solving

\* Creative Thinking

\* Productivity

- Category features:

\* Browse by category

\* Skill progression

\* Difficulty levels

\* Related skills

## Technical Requirements

### Frontend (Flutter)

- Dependencies:

\* Flutter SDK (latest stable)

\* GetX state management

\* Better Player

\* Firebase SDK

- Performance metrics:

\* Launch time < 2s

\* Smooth scrolling (60fps)

\* Video load time < 1s

\* Offline support

### Backend (Firebase)

- Services:

\* Authentication

\* Cloud Firestore

\* Cloud Storage

\* Cloud Functions

\* Cloud Messaging

- Database collections:

\* Users

\* Videos

\* Skills

\* Progress

### Video Processing

- Processing features:

\* Automatic compression

\* Thumbnail generation

\* Quality optimization

\* Format standardization

- Delivery features:

\* CDN distribution

\* Adaptive streaming

\* Cache management

## Project Structure

lib/

├── core/

│ ├── constants/

│ ├── theme/

│ └── utils/

├── features/

│ ├── auth/

│ ├── feed/

│ ├── profile/

│ └── skills/

├── models/

├── services/

└── widgets/

## Database Schema

- users/

\* profile

\* skills

\* progress

\* saved\_videos

- videos/

\* metadata

\* engagement

\* verification

\* categories

- skills/

\* categories

\* progression

\* challenges

## MVP Timeline (Week 1)

### Day 1-2

- Project setup

- Firebase configuration

- Authentication implementation

- Basic feed structure

### Day 3-4

- Video player integration

- User interactions

- Progress tracking system

- Basic UI components

### Day 5

- Testing

- Bug fixes

- Documentation

- Deployment preparation

## Success Metrics

- Key metrics:

\* Daily Active Users

\* Average Watch Time

\* Completion Rates

\* Return Rate

\* Share Rate

- Technical metrics:

\* Video load success > 98%

\* App crash rate < 0.1%

\* User retention D1 > 40%

## Development Guidelines

1. Follow Flutter best practices

2. Implement clean architecture

3. Use GetX for state management

4. Follow atomic design for UI

5. Implement comprehensive error handling

6. Ensure accessibility compliance

7. Maintain offline functionality

8. Optimize for performance

## Testing Requirements

1. Unit tests for business logic

2. Widget tests for UI components

3. Integration tests for user flows

4. Performance testing

5. Security testing

6. User acceptance testing

## Deployment Requirements

1. CI/CD pipeline setup

2. App store optimization

3. Version management

4. Crash reporting

5. Analytics integration

6. Monitoring setup

## Essential Dependencies

yaml

dependencies:

flutter:

sdk: flutter

get: ^4.6.5

firebase\_core: ^2.4.1

firebase\_auth: ^4.2.5

cloud\_firestore: ^4.3.1

firebase\_storage: ^11.0.10

better\_player: ^0.0.83

cached\_network\_image: ^3.2.3

image\_picker: ^0.8.6

video\_compress: ^3.1.2

shared\_preferences: ^2.0.17

flutter\_svg: ^1.1.6

lottie: ^2.2.0

## Future Considerations (Week 2)

- AI features:

\* Personalized recommendations

\* Skill path generation

\* Progress prediction

\* Content verification

\* Engagement optimization

- Community features:

\* Challenges

\* User achievements

\* Skill validation

\* Expert verification

## Performance Requirements

1. App size < 50MB

2. Cold start < 2 seconds

3. Video start time < 1 second

4. Smooth scrolling (60fps)

5. Offline functionality

6. Battery efficient

7. Data efficient

## Security Requirements

1. Secure user authentication

2. Data encryption

3. Safe content storage

4. API security

5. Content moderation

6. Privacy compliance

## Analytics Requirements

1. User engagement metrics

2. Video performance metrics

3. Learning progress metrics

4. Error tracking

5. Performance monitoring

6. User behavior analytics

Note: This PRD is designed to be fed into Cursor agent for development assistance and code generation. All technical specifications should be reviewed and adjusted based on specific implementation requirements.