JORTS (Join Others, Rate Today's Styles)

JORTS is a unique social media platform combining the fun of sharing your daily outfits with the interactive twist of “Would You Rather.” Users are prompted to upload a daily outfit photo ("fit pic") and then participate in a game-like environment where they vote on other users' outfits. The platform enables users to share details about their clothing when posting their outfits, add reactions, and receive feedback from others, fostering a fashion-focused community.

Links

- A clear and detailed list of commits showing the contributions of each member over time can be found on the group’s Github Repository
  - https://github.com/csc-4330-amin-duc-kai-riley/4330-group-project
- Github repository “main” branch
  - This branch contains all of the project documentation for class assignments related to the project such as milestone 1, milestone 2, milestone3, etc.
    - https://github.com/csc-4330-amin-duc-kai-riley/4330-group-project/tree/main
- Github repository “jort_code” branch
  - This branch contains the complete implementation of the project's source code.
    - https://github.com/csc-4330-amin-duc-kai-riley/4330-group-project/tree/jorts_code
- Website Video Demo
  - https://youtu.be/ZMXvxFSj7XU
- Note: Other branches in the GitHub repository, such as "Amin," "jorts_backend," and "master," were used for testing and prototyping purposes. These branches are not part of the final implementation.
- Google Slides Final Presentation
  - https://docs.google.com/presentation/d/1UiytQBHjD9ieRPBzMOzkW9ViOUljj8T3qYKiepj5iVg/edit#slide=id.g31b2a0f57aa_1_5

Core Features

User Authentication

- Secure user registration and login system
- JWT-based authentication
- Protected routes for authenticated users
- Profile customization options

Daily Outfit Uploads

- Daily outfit photo submissions
- Detailed outfit information including:
  - Item descriptions
  - Brand tags
  - Shopping links
- Image storage via Cloudinary integration

Interactive Voting System

- Head-to-head outfit comparisons
- Real-time vote tracking
- Daily voting pairs
- Vote statistics and analytics

Profile Management

- Customizable user profiles
- Outfit history tracking
- Following/Follower system
- Activity statistics

Social Features

- Comment system
- Outfit reactions
- User interactions
- Community engagement metrics

Technical Implementation

Frontend

- React.js with Material Dashboard 2 template
- Material-UI components
- Responsive design
- Protected/public route management
- JWT authentication handling

Backend

- Node.js/Express server
- MongoDB Atlas database
- RESTful API architecture
- Cloudinary integration for images
- Secure authentication middleware

Database Structure

- Users collection (profiles, credentials)
- Posts collection (outfits, metadata)
- PostPairs collection (voting pairs)
- Votes collection (voting records)
- Comments collection (user interactions)

Usage

1. Create an account or use test credentials
2. Upload your daily outfit with details
3. Vote on other users' outfit pairs
4. Interact with the community through comments and reactions
5. Track your style journey through your profile statistics

JORTS Installation Guide

Prerequisites

1. Node.js (v14 or higher)

- Download from: https://nodejs.org/
- Verify installation:

```
node --version
```

2. MongoDB Community Edition

Mac OS Installation:

Using Homebrew

```
brew tap mongodb/brew brew install mongodb-community@7.0
```

Start MongoDB service

```
brew services start mongodb-community
```

Windows Installation:

1. Download MongoDB Community Server from https://www.mongodb.com/try/download/community
2. Run the installer (.msi file)
3. Follow installation wizard (select "Complete" installation)
4. Start MongoDB service:

In Windows PowerShell as Administrator

```
net start MongoDB
```

3. Git

- Download from: https://git-scm.com/
- Verify installation:

```
git --version
```

Project Setup

1. Clone Repository

```
git clone [your-repository-url]
```

```
cd jorts_code
```

2. Backend Setup

Navigate to backend directory

```
cd backend
```

Install dependencies

```
npm install express mongoose jsonwebtoken bcryptjs cors dotenv cloudinary multer
```

3. Frontend Setup

Navigate back to main project directory

```
cd jorts_code
```

Install frontend dependencies

```
npm install
```

4. Environment Setup

Create a file named .env in the backend directory with the following content:

```
MONGODB_URI=mongodb+srv://[your-connection-string]
JWT_SECRET=jorts_secret_key_123456789 PORT=3001 NODE_ENV=development
```

Running the Application

1. Start MongoDB

Mac OS: brew services start mongodb-community

Windows:

In Windows PowerShell as Administrator

```
net start MongoDB
```

2. Start Backend Server

In backend directory

```
npm run dev
```

3. Start Frontend Development Server

In main project directory

```
npm start
```

The application should now be running at:

- Frontend: http://localhost:3000
- Backend: http://localhost:3001

Test Setup

Create Test Data

In backend directory

```
node scripts/seedDemoData.js
```

Test User Credentials

- Username: testuser
- Password: testuser707