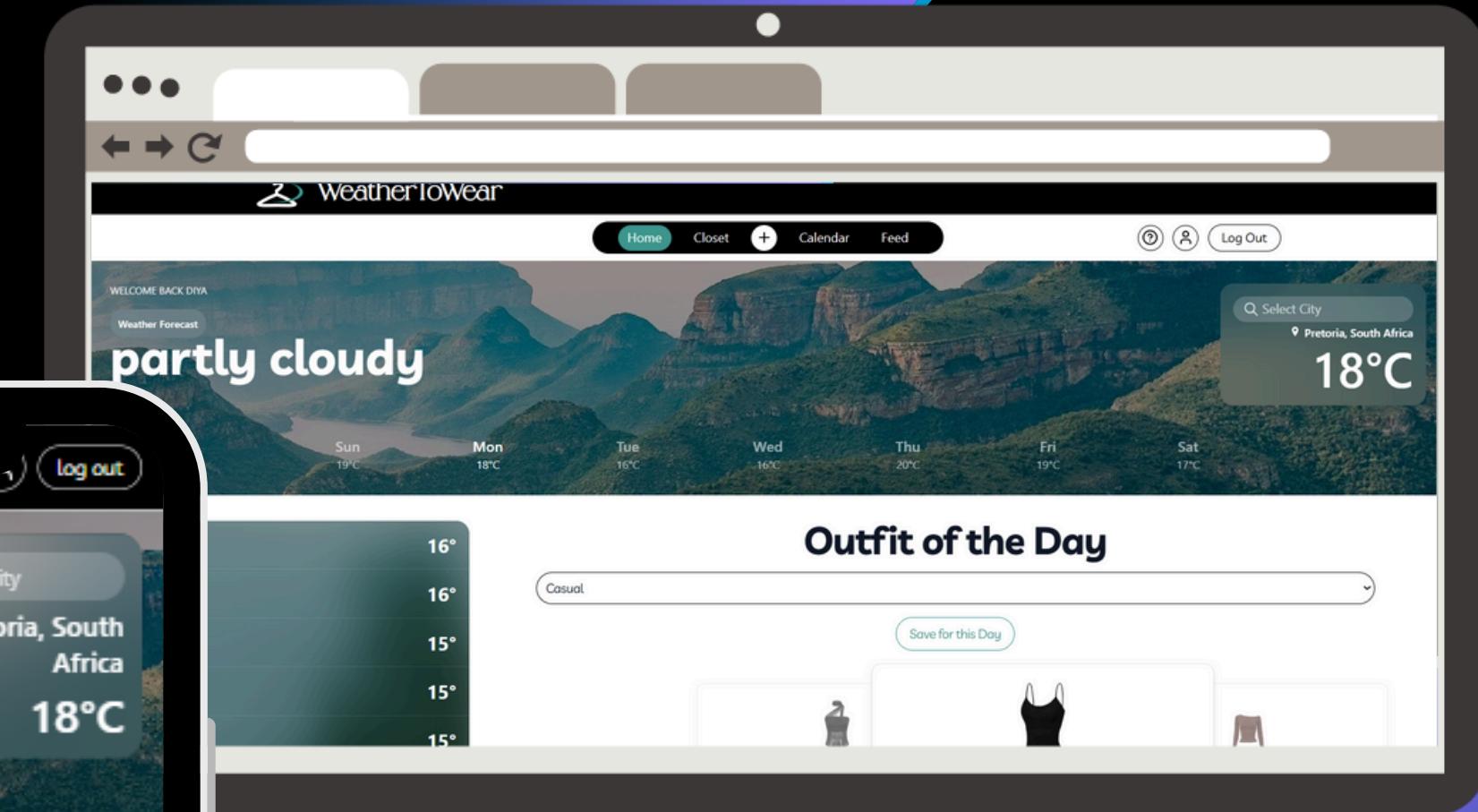
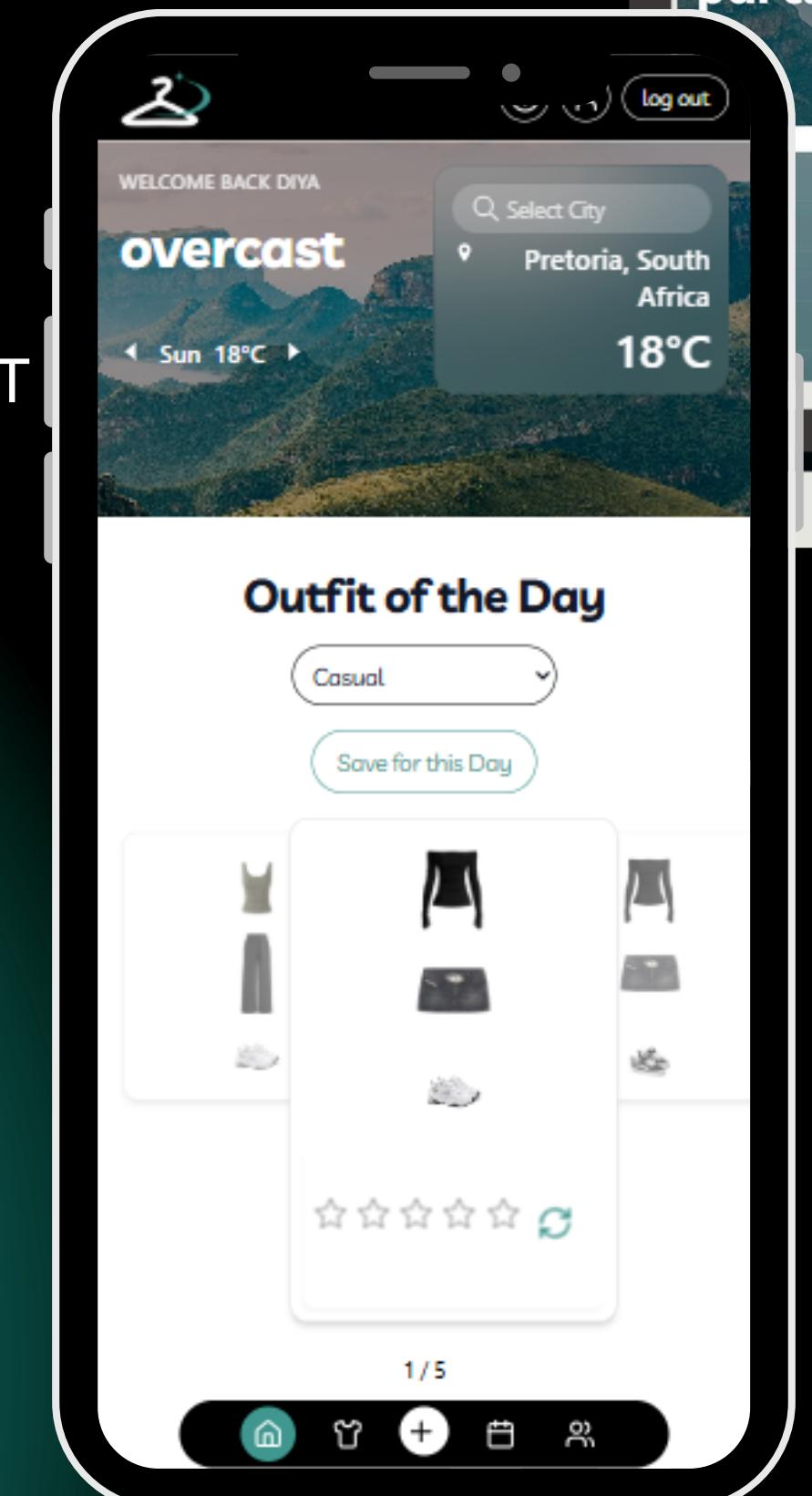


WeatherTo Wear

AN EPI USE AFRICA PROJECT



GitGood

01 The Problem

Millions of people face the same challenge.
What to wear? It sounds simple, but it's actually
stressful and time-consuming.

Outfit of the Day

Casual

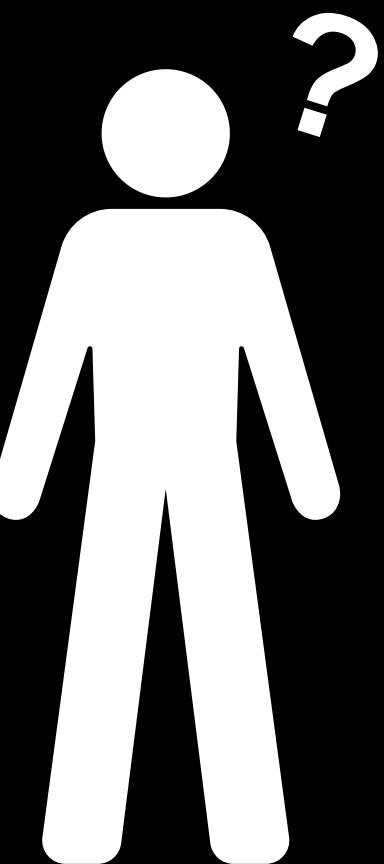
Save for this Day



★★★★★



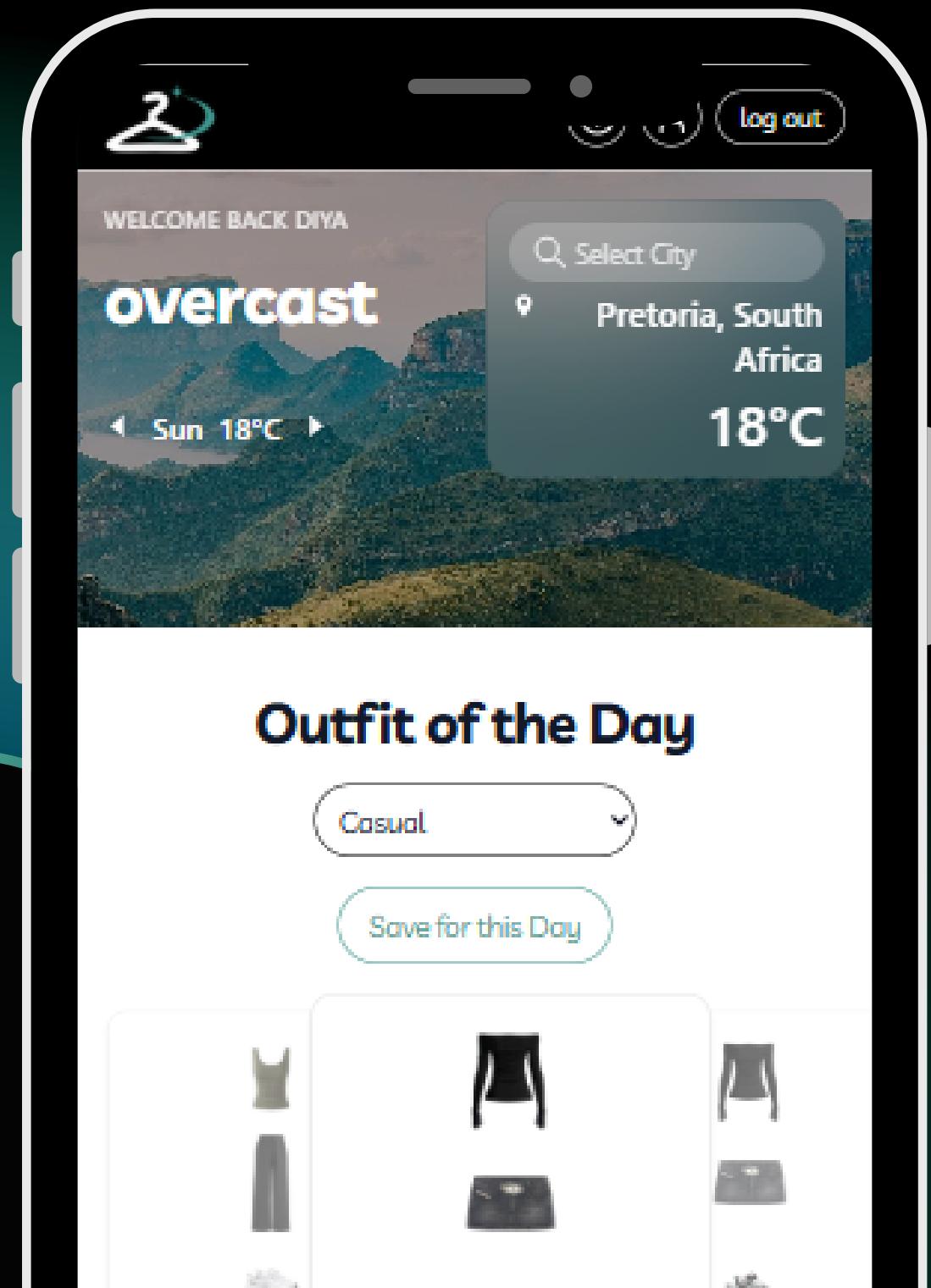
Going to rain?



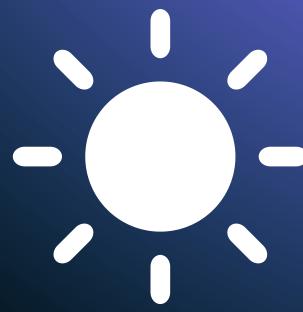
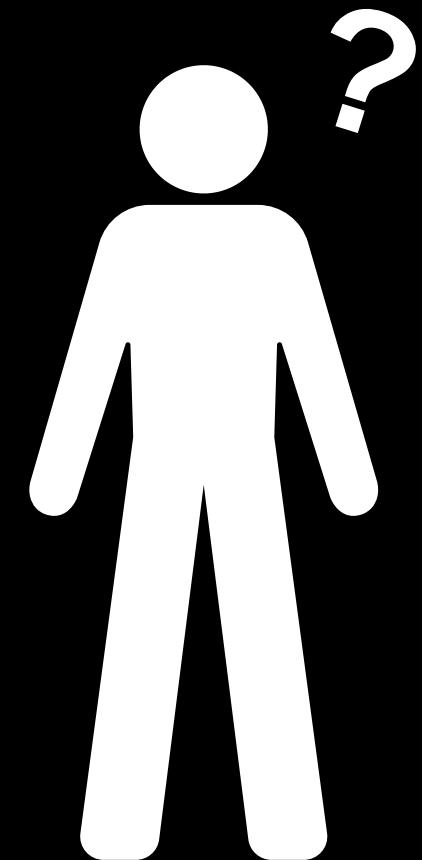
Too hot for a jacket?

01 The Problem

Millions of people face the same challenge.
What to wear. It sounds simple, but it's actually
stressful and time-consuming.



Going to rain?



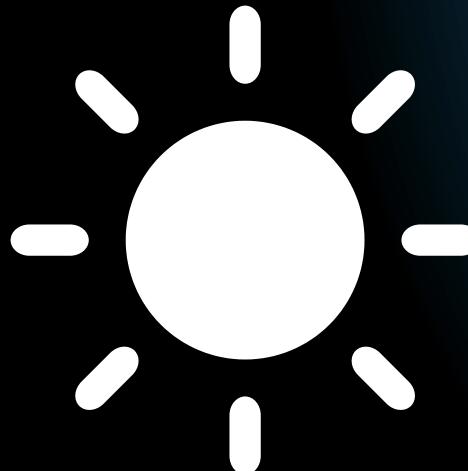
Too hot for a jacket?

01 The Problem

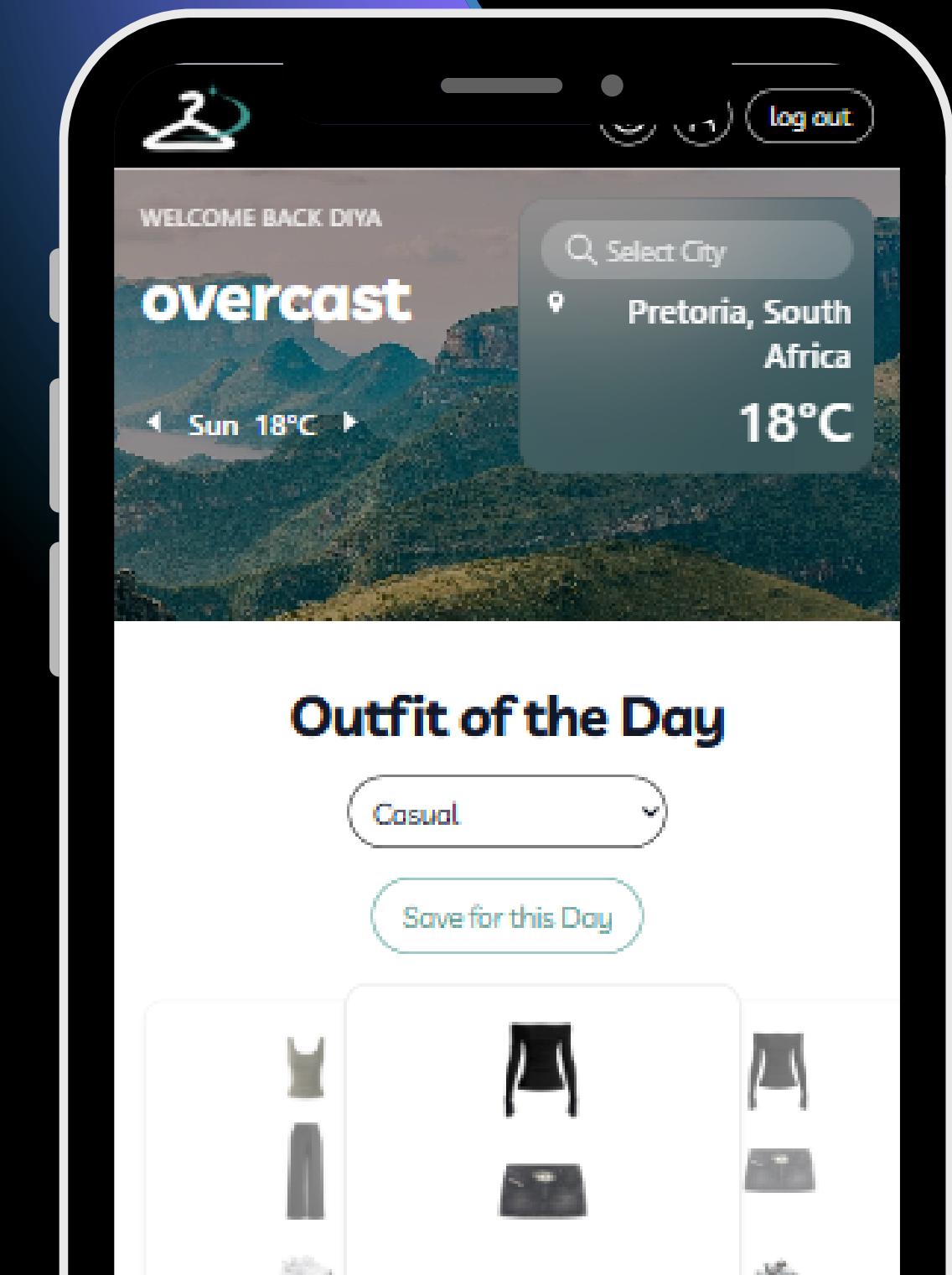
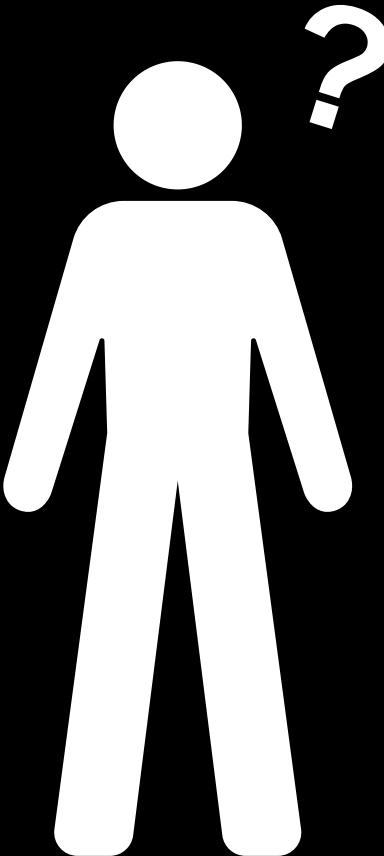
Millions of people face the same challenge.
What to wear. It sounds simple, but it's actually
stressful and time-consuming.



Going to rain?

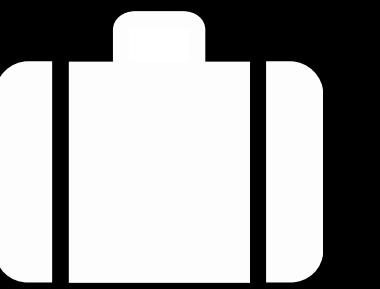


Too hot for a jacket?



02 The Problem

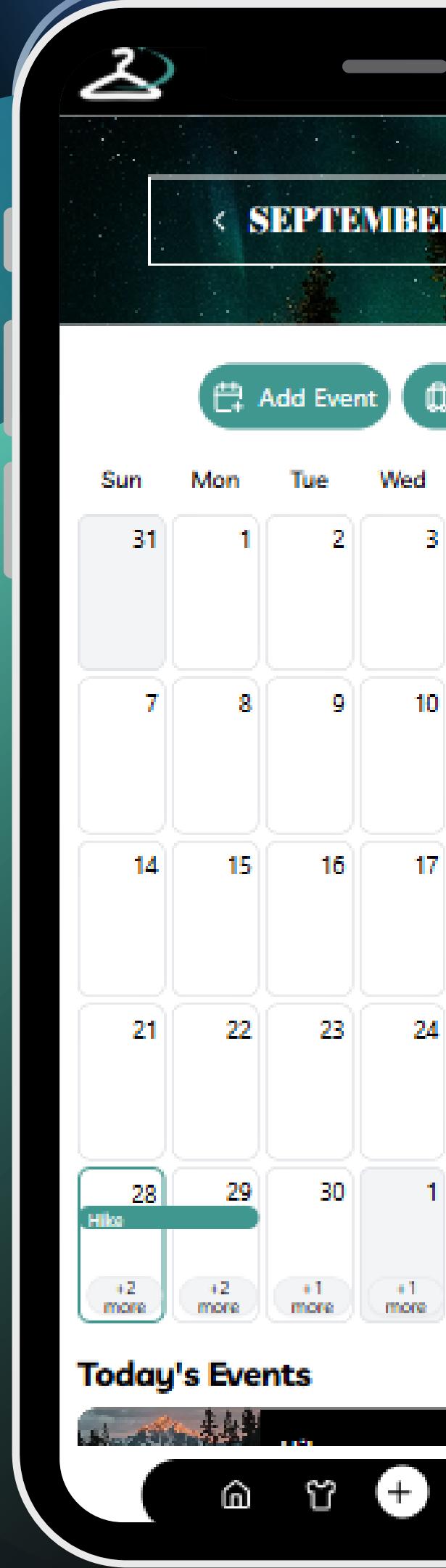
The reoccurring struggle of picking the perfect outfit for that special occasion



What to pack for your trip

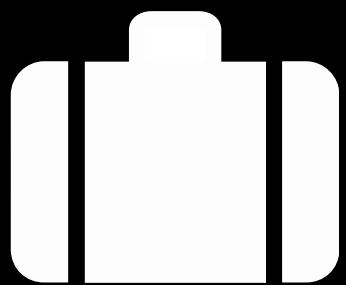


Perfect date outfit

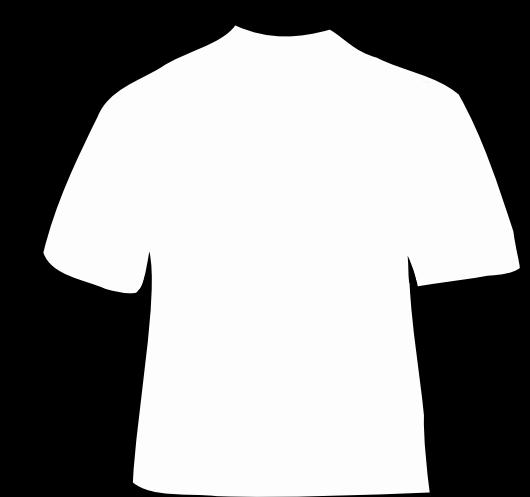


02 The Problem

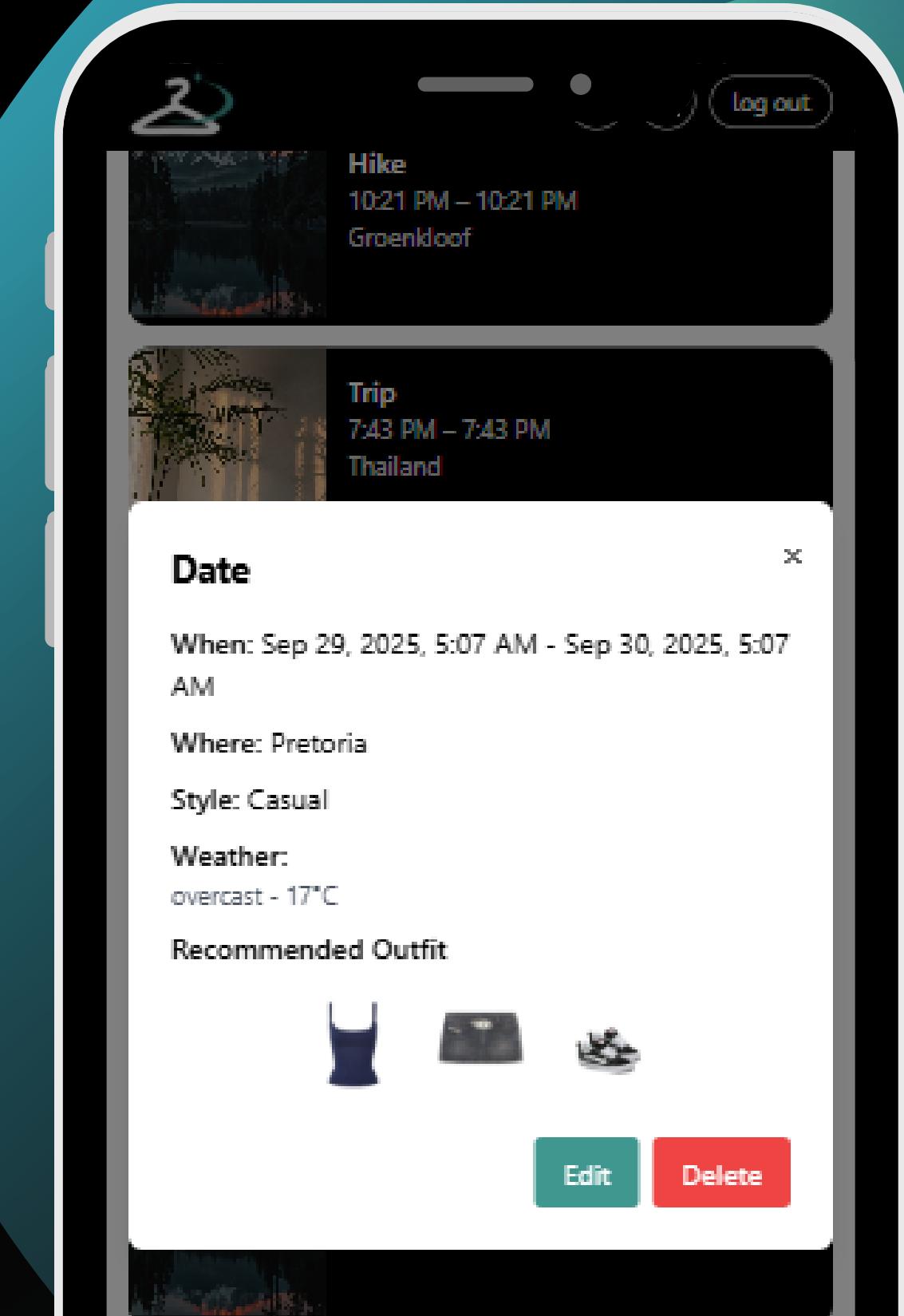
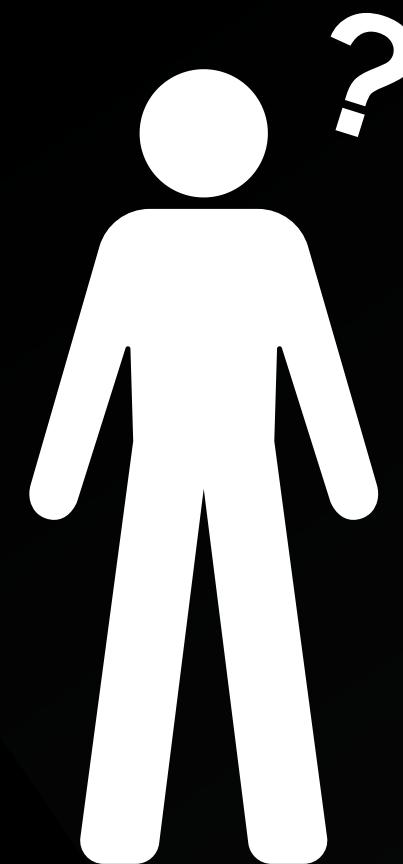
The reoccurring struggle of picking the perfect outfit for that special occasion



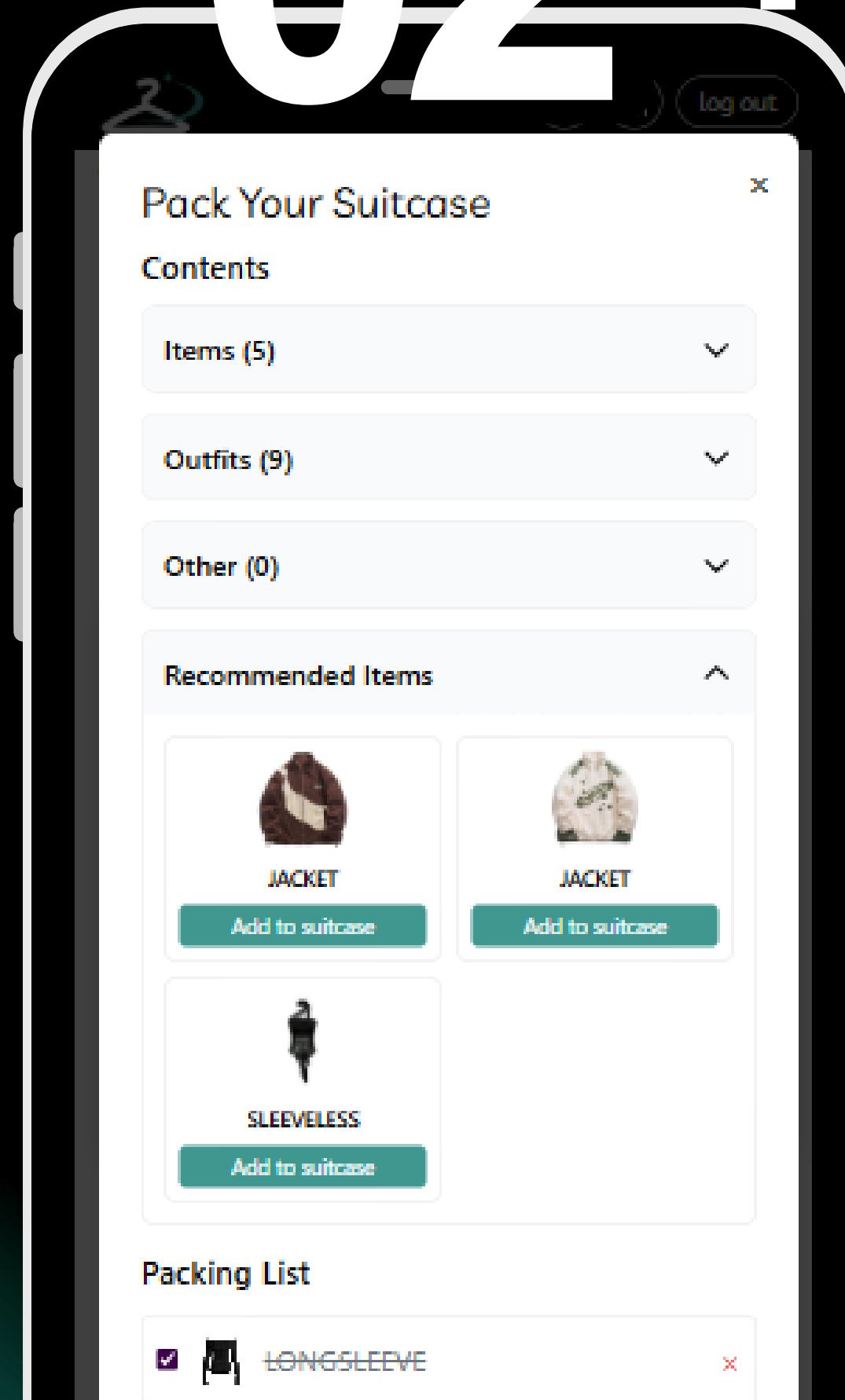
What to pack for your trip



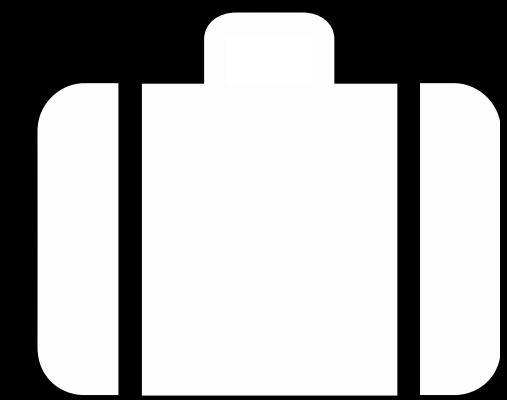
Perfect date outfit



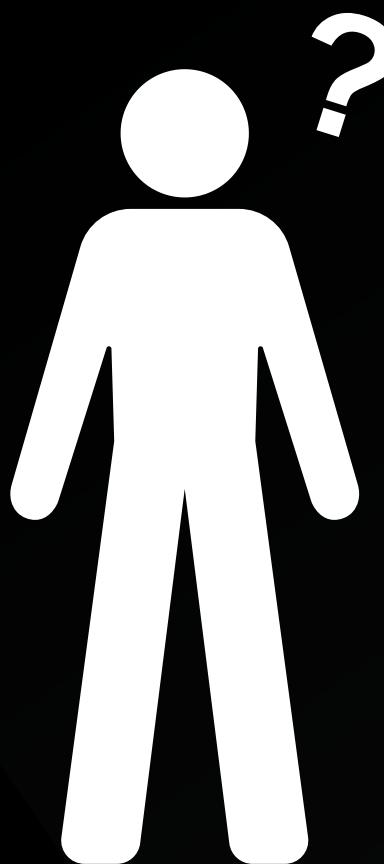
02 The Problem



The reoccurring struggle of picking the perfect outfit for that special occasion



What to pack for your trip

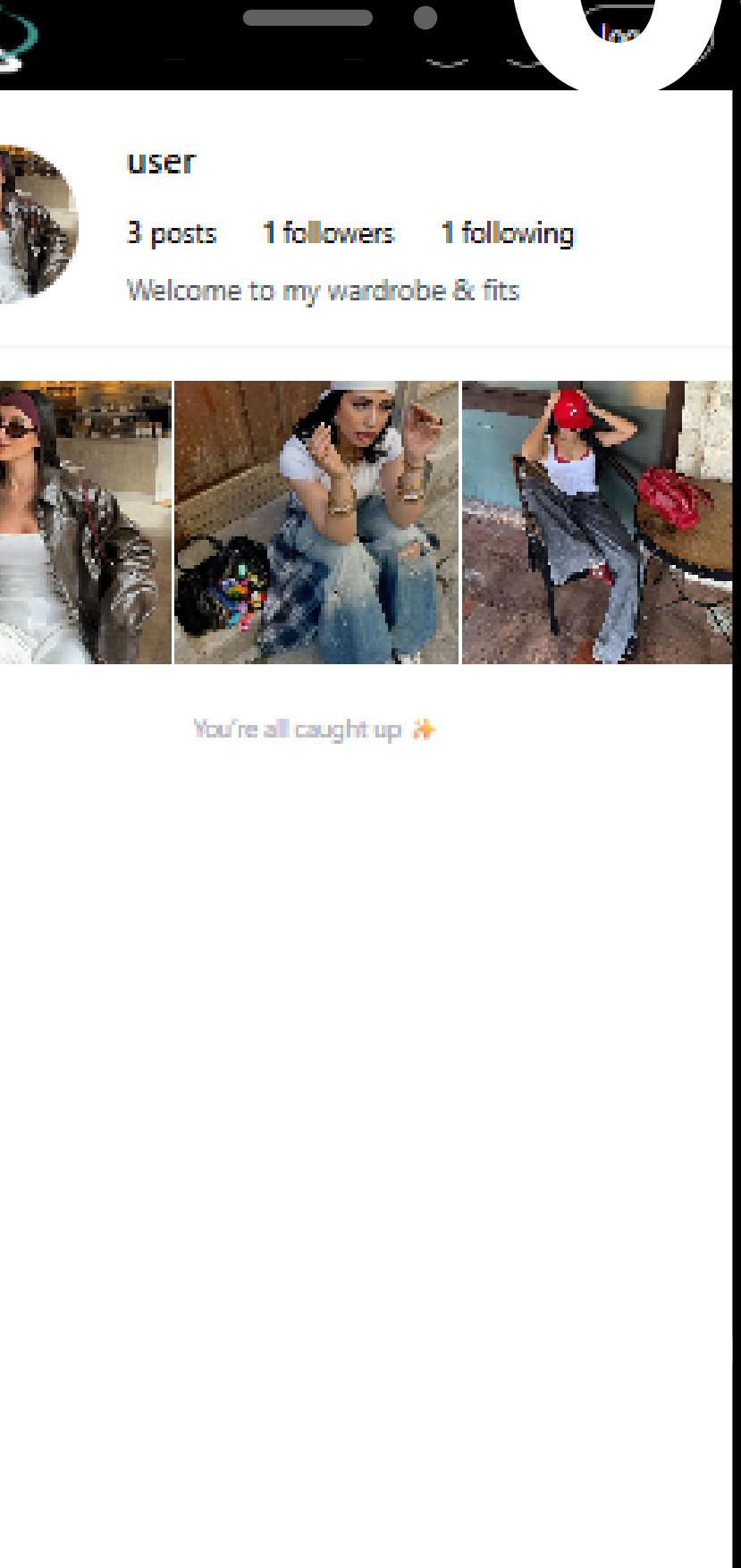


Perfect date outfit

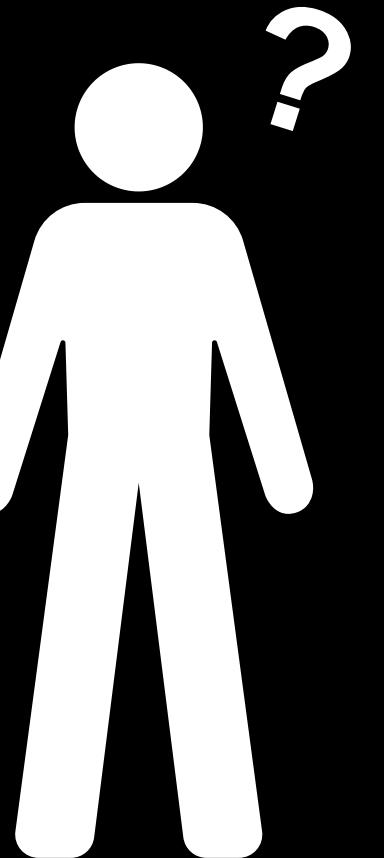
03

The Problem

Keeping up with trending looks and what
your friends are rocking



Want to match with your friends?

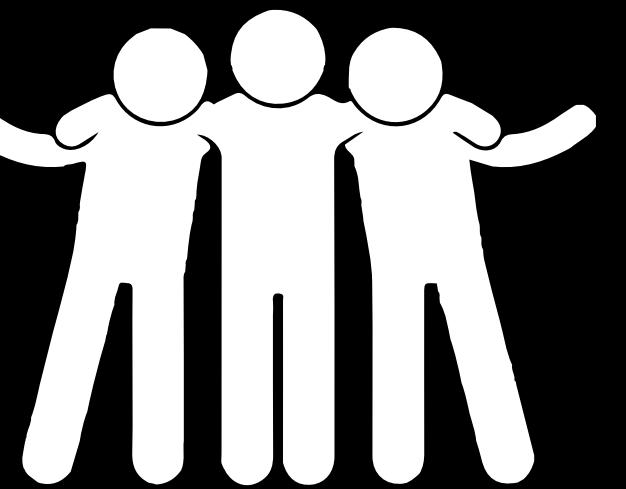


*What to stay on top of the
fashion game*

03 The Problem



Keeping up with trending looks and what
your friends are rocking



Want to match with your friends?



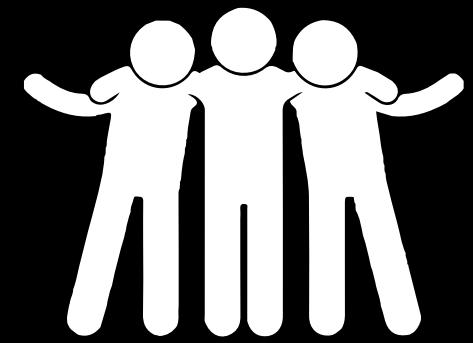
*What to stay on top of the
fashion game*

03 The Problem

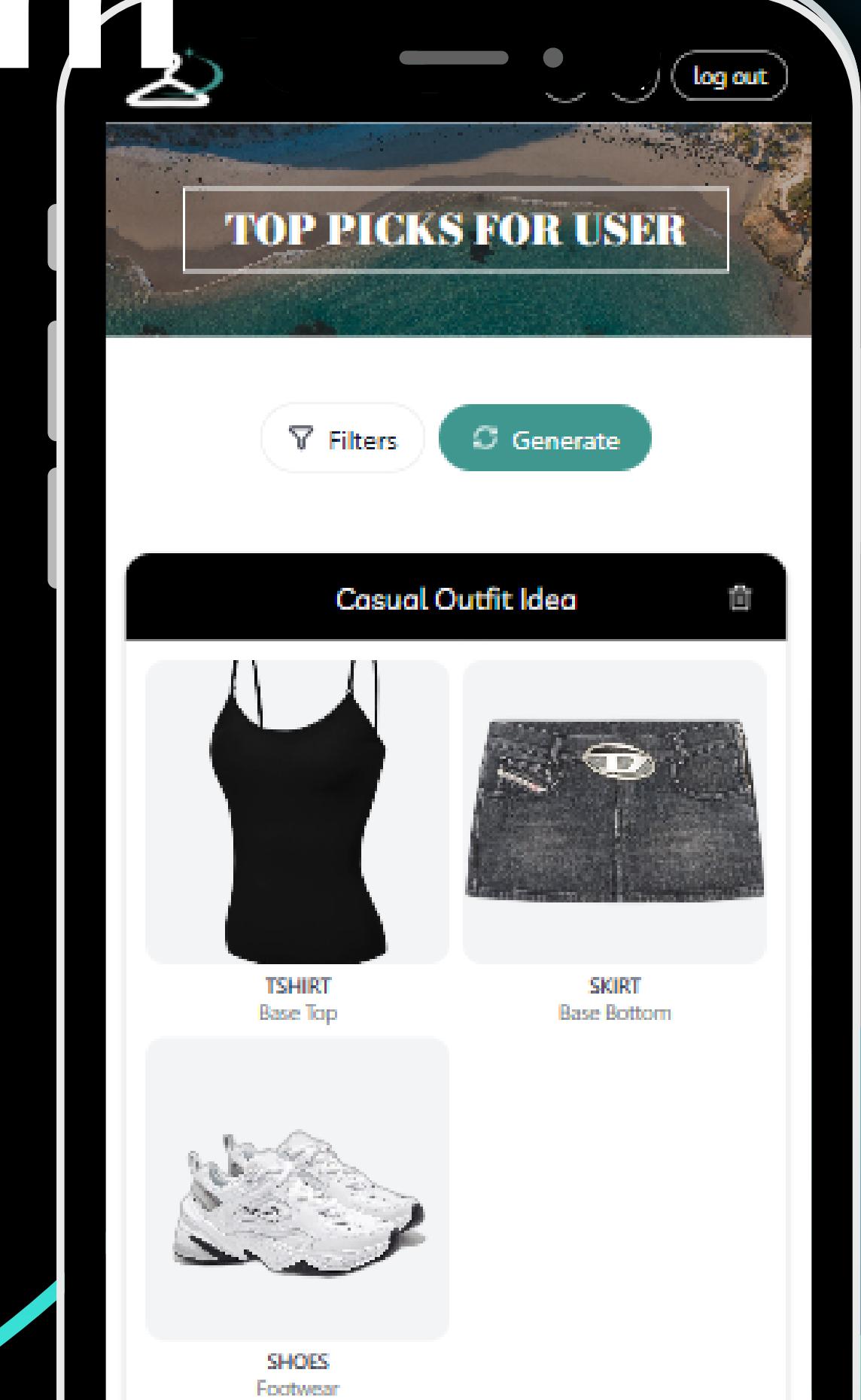
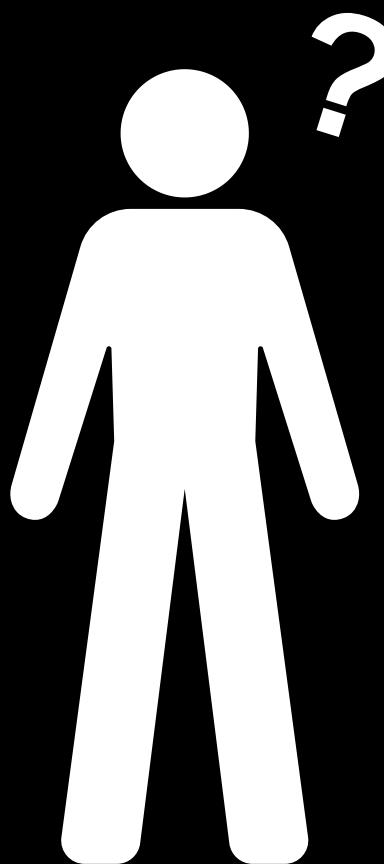
Keeping up with trending looks and what
your friends are rocking



*What to stay on top of the
fashion game*



Want to match with your friends?

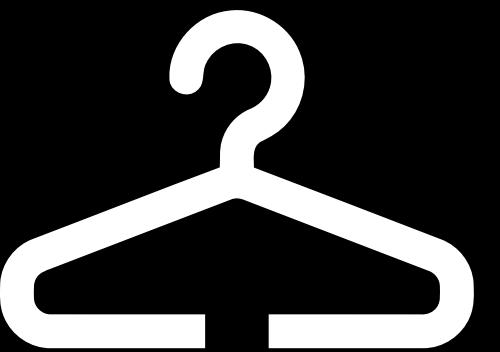


04 The Problem

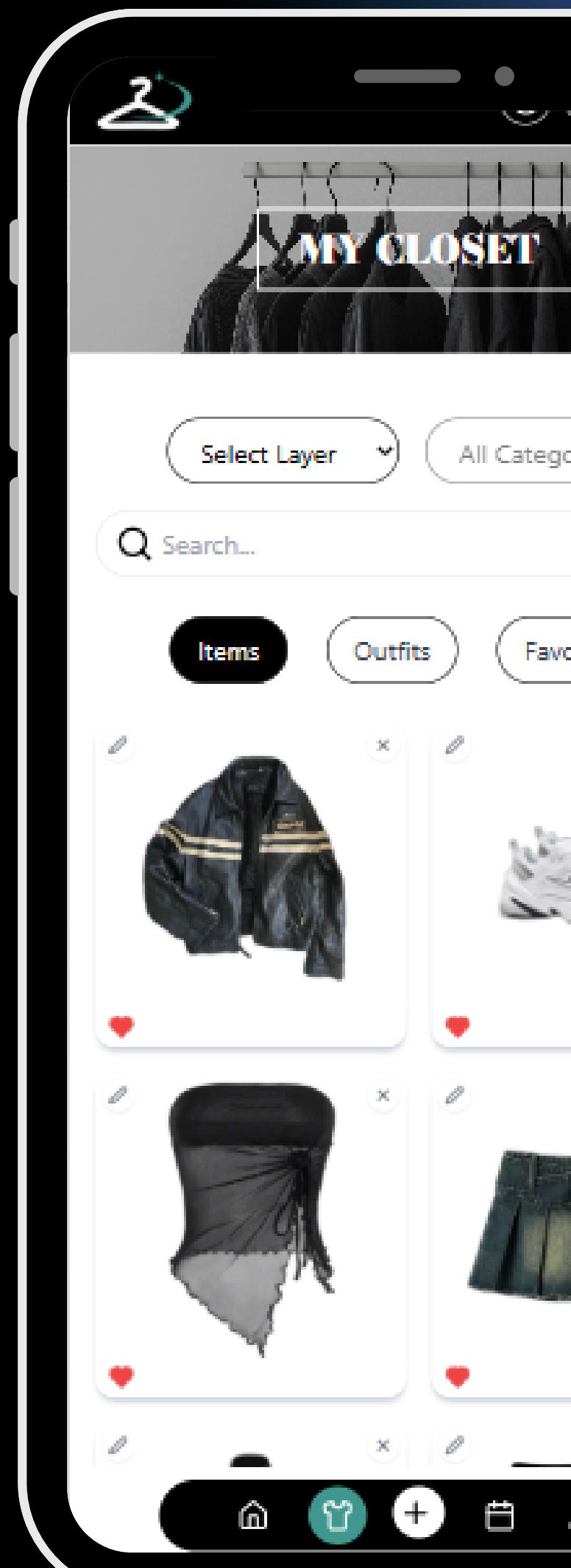
Tearing your closet apart trying on different outfits



Want to see your closet in seconds

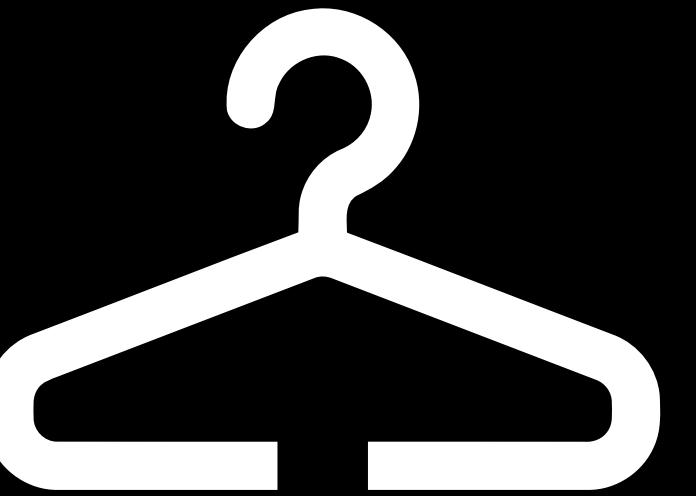


Try on outfits while in bed



04 The Problem

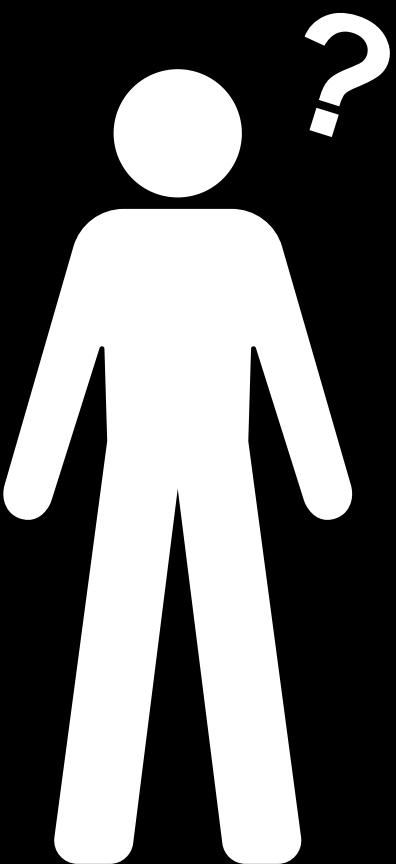
Tearing your closet apart trying on different outfits



Try on outfits while in bed

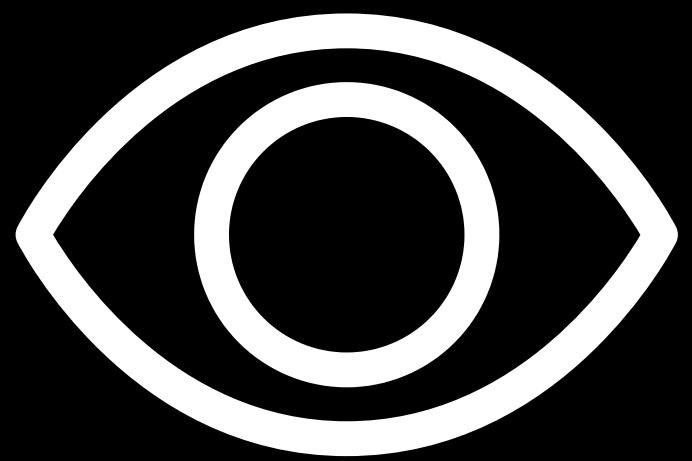
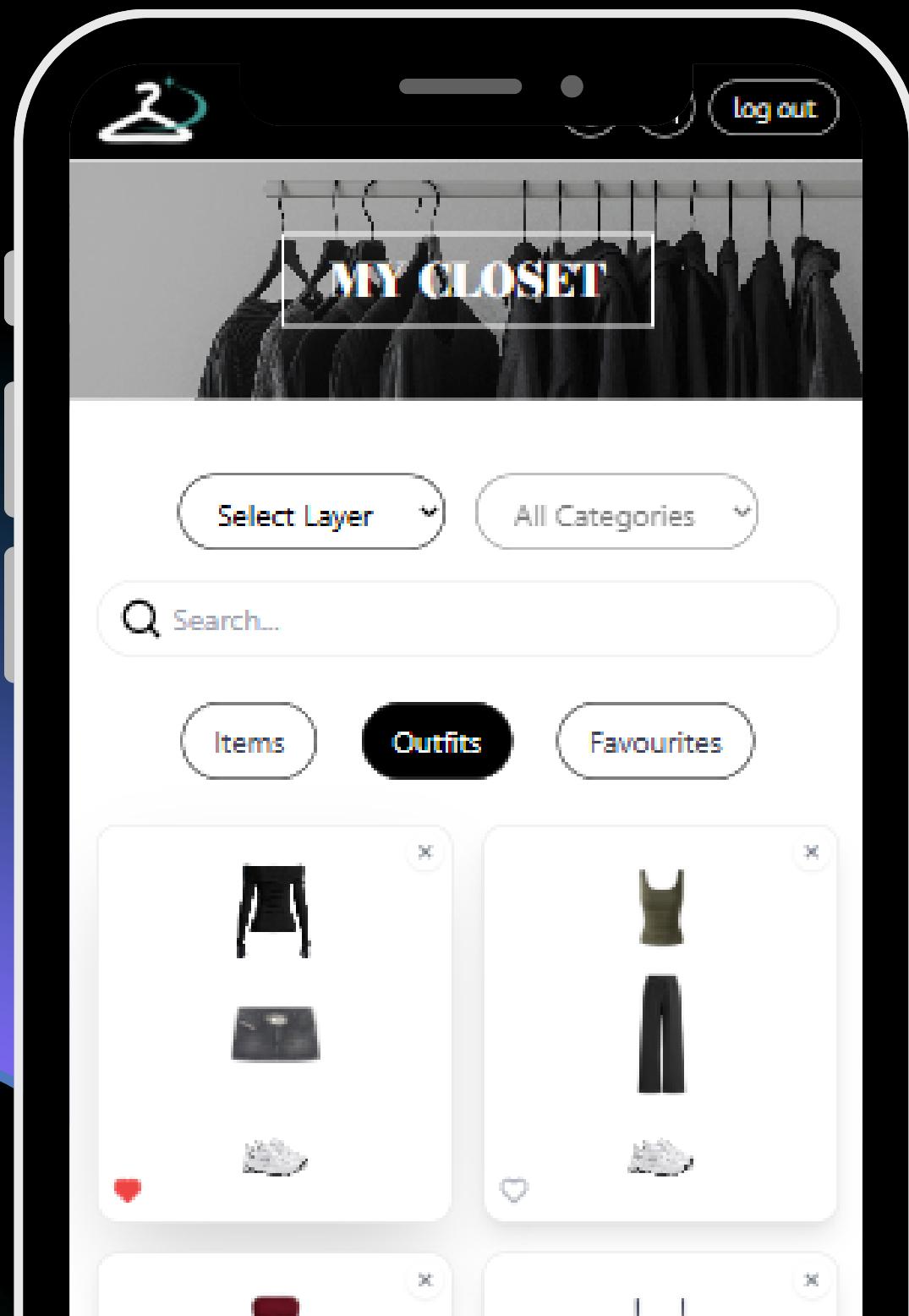


Want to see your closet in seconds

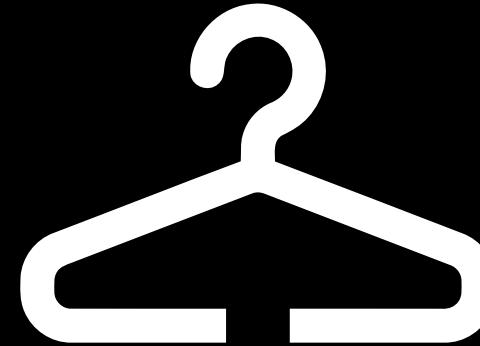
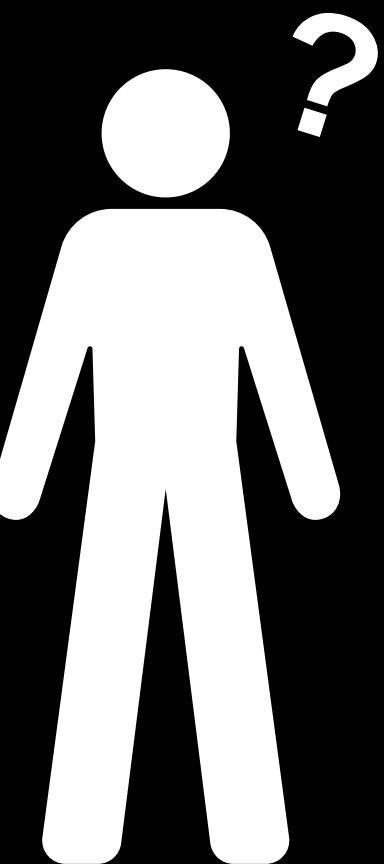


04 The Problem

Tearing your closet apart trying on different outfits



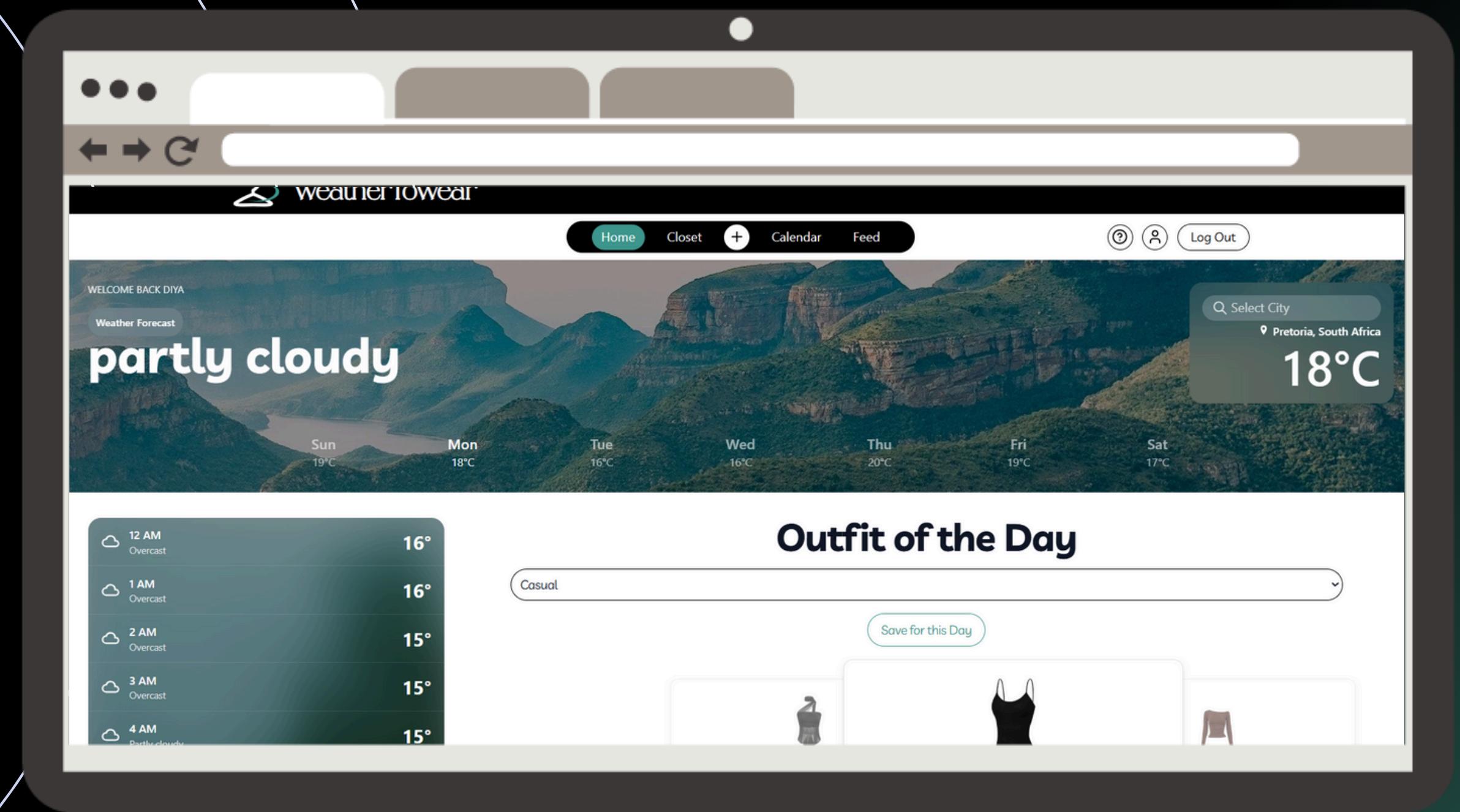
Want to see your closet in seconds



Try on outfits while in bed

Live Demo

Core Functionality and Wow Factors



01 Recommendation Engine

Rule-Based Engine

1. Fetch outfits
2. Decide which layers make sense today
3. Adjust to the weather conditions
4. Build candidate outfits
5. Score each outfit depending on:
 - a. Colour harmony
 - b. User's preferred colours
 - c. Warmth
 - d. Rain bonus

02 Recommendation Engine

K Nearest Neighbours



1. Turn outfit into numbers
2. Similarity is measured (Cosine Similarity)
3. Predict a rating from neighbours

03 Recommendation Engine

Collaborative Filtering

1. Build a taste profile per user
2. Find users like 'me'
3. Predict how my neighbours would rate this outfit
4. Blend with other models

Non-functional Requirements

01 Performance

02 Scalability

03 Security

04 Usability

Non-functional Requirements Testing

01 Performance Smoke

```
execution: local
script: /scripts/smoke-read.k6.js
output: -  
  
scenarios: (100.00%) 1 scenario, 5 max VUs, 1m30s max duration (incl. graceful stop):
* default: 5 looping VUs for 1m0s (gracefulStop: 30s)  
  
THRESHOLDS  
  
http_req_duration{endpoint:closet_all}
✓ 'p(95)<800' p(95)=222.34ms
✓ 'p(99)<1500' p(99)=240.74ms  
  
http_req_duration{endpoint:outfits_recommend}
✓ 'p(95)<1500' p(95)=441.04ms  
  
http_req_duration{endpoint:weather}
✓ 'p(95)<3500' p(95)=3.2s
✓ 'p(99)<4000' p(99)=3.22s  
  
http_req_failed
✓ 'rate<0.01' rate=0.00%  
  
TOTAL RESULTS  
  
checks_total.....: 546    8.540183/s
checks_succeeded.: 100.00% 546 out of 546
checks_failed....: 0.00%   0 out of 546  
  
✓ weather 200
✓ weather repeat 200
✓ closet 200
✓ recommend 200
✓ recommend returns array
✓ recommend <=1.5s  
  
CUSTOM
weather_first_ms.....: avg=232.065934 min=183      med=190      max=842      p(90)=229      p(95)=742
weather_repeat_ms.....: avg=2160.791209 min=199      med=3188     max=3759     p(90)=3209     p(95)=3221.5  
  
HTTP
http_req_duration.....: avg=710.78ms  min=182.93ms  med=212.4ms  max=3.35s   p(90)=3.19s   p(95)=3.19s
  { endpoint:closet_all }.....: avg=197.73ms  min=187.12ms  med=194.79ms  max=242.51ms  p(90)=212.26ms  p(95)=222.34ms
  { endpoint:outfits_recommend }: avg=308.94ms  min=271.35ms  med=279.36ms  max=765.55ms  p(90)=366.38ms  p(95)=441.04ms
  { endpoint:weather }.....: avg=1.17s    min=182.93ms  med=204.27ms  max=3.35s   p(90)=3.19s   p(95)=3.2s
  { expected_response:true }...: avg=710.78ms  min=182.93ms  med=212.4ms  max=3.35s   p(90)=3.19s   p(95)=3.19s
http_req_failed.....: 0.00%   0 out of 365
http_reqs.....: 365    5.709097/s  
  
EXECUTION
iteration_duration...: avg=3.43s      min=1.11s      med=4.19s      max=5.05s      p(90)=4.9s      p(95)=4.95s
iterations.....: 91    1.423364/s
vus.....: 3      min=3      max=5
vus_max.....: 5      min=5      max=5  
  
NETWORK
data_received.....: 2.8 MB 43 kB/s
data_sent.....: 194 kB 3.0 kB/s  
  
Running (1m03.9s), 0/5 VUs, 91 complete and 0 interrupted iterations
default ✓ [=====] 5 VUs 1m0s
S C:\Users\kylea\Documents\w2w\Weather-to-Wear\app-backend> |
```

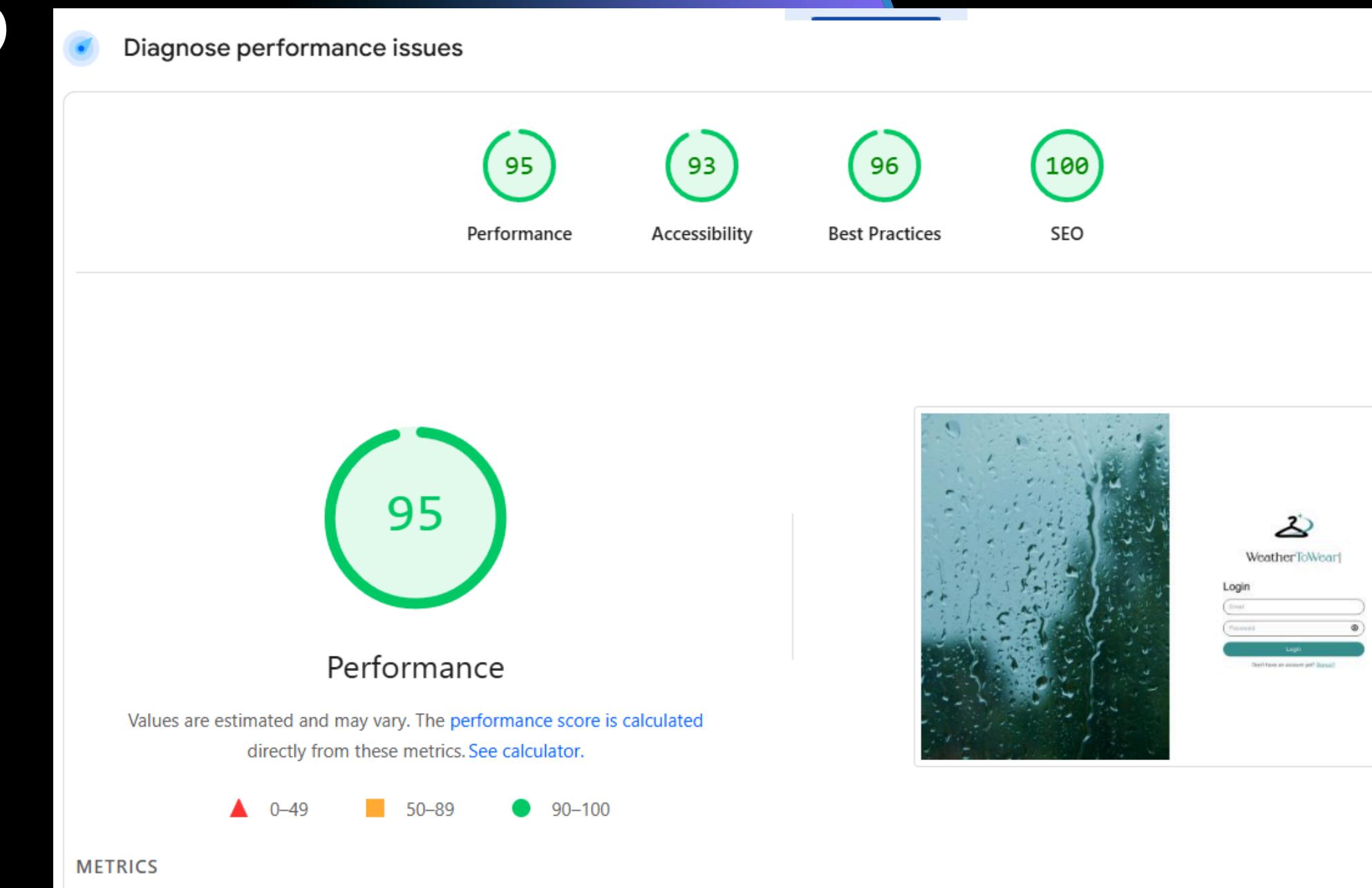
Non-functional Requirements Testing

01 Performance Pipeline

```
execution: local
script: upload-pipeline.k6.js
output: -  
  
scenarios: (100.00%) 1 scenario, 1 max VUs, 10m30s max duration (incl. graceful stop):
  * default: 3 iterations shared among 1 VUs (maxDuration: 10m0s, gracefulStop: 30s)  
  
■ THRESHOLDS  
  
e2e_total_ms
✓ 'p(50)<9000' p(50)=8372
✓ 'p(95)<12000' p(95)=8739.2  
  
http_req_failed
✓ 'rate<0.05' rate=0.00%  
  
pipeline_total_ms
✓ 'p(50)<7000' p(50)=752
✓ 'p(95)<10000' p(95)=780.8  
  
■ TOTAL RESULTS  
  
checks_total.....: 12      0.43589/s
checks_succeeded.: 100.00% 12 out of 12
checks_failed....: 0.00%   0 out of 12  
  
✓ upload 201
✓ upload returns id
✓ upload returned id
✓ pipeline finished <=10s  
  
CUSTOM
e2e_total_ms.....: avg=8450      min=8198      med=8372      max=8780      p(90)=8698.4      p(95)=8739.2
pipeline_total_ms.: avg=759       min=741       med=752       max=784       p(90)=777.6      p(95)=780.8
upload_latency_ms.: avg=7690.666667 min=7446      med=7587      max=8039      p(90)=7948.6      p(95)=7993.8  
  
HTTP
http_req_duration...: avg=3.34s      min=192.05ms med=278.24ms max=7.58s p(90)=7.52s  p(95)=7.55s
  { expected_response:true }.: avg=3.34s      min=192.05ms med=278.24ms max=7.58s p(90)=7.52s  p(95)=7.55s
http_req_failed.....: 0.00%   0 out of 7
http_reqs.....: 7      0.254269/s  
  
EXECUTION
iteration_duration...: avg=8.87s      min=8.59s      med=8.77s      max=9.26s      p(90)=9.16s  p(95)=9.21s
iterations.....: 3      0.108972/s
vus.....: 1      min=1      max=1
vus_max.....: 1      min=1      max=1  
  
NETWORK
data_received.....: 57 kB 2.1 kB/s
data_sent.....: 90 kB 3.3 kB/s  
  
running (00m27.5s), 0/1 VUs, 3 complete and 0 interrupted iterations
default ✓ [=====] 1 VUs  00m26.6s/10m0s  3/3 shared iters
PS C:\Users\kylea\Documents\w2w\Weather-to-Wear\app-backend> |
```

Non-functional Requirements Testing

01 Performance Lighthouse



Non-functional Requirements Testing

01 Performance SLOs

- API reads p95 ≤ 800 ms (p99 ≤ 1500 ms) @ 20 req/s
- 5-outfit generation ≤ 1.5 s with 50 items
- BG removal median ≤ 5 s; colour extraction median ≤ 2 s.
- API for weather reads p95 ≤ 3500 ms (p99 ≤ 4000 ms) @ 20 req/s

Non-functional Requirements Testing

02 Scalability

Scale Read

```
execution: local
script: scale-read.k6.js
output: -  
  
scenarios: (100.00%) 3 scenarios, 44 max VUs, 5m30s max duration (incl. graceful stop):
* closet: 3.00 iterations/s for 5m0s (maxVUs: 12-24, exec: closet, gracefulStop: 30s)
* recommend: 1.00 iterations/s for 5m0s (maxVUs: 5-10, exec: recommend, gracefulStop: 30s)
* weather: 1.00 iterations/s for 5m0s (maxVUs: 5-10, exec: weather, gracefulStop: 30s)
```

THRESHOLDS

```
http_req_duration{endpoint:closet_all}
✓ 'p(95)<400' p(95)=224.41ms
✓ 'p(99)<900' p(99)=263.9ms  
  
http_req_duration{endpoint:outfits_recommend}
✓ 'p(95)<1000' p(95)=341.91ms  
  
http_req_duration{endpoint:weather}
✓ 'p(95)<3500' p(95)=3.22s  
  
http_req_failed
✓ 'rate<0.01' rate=0.00%
```

TOTAL RESULTS

```
checks_total.....: 1503    4.95854/s
checks_succeeded.: 100.00% 1503 out of 1503
checks_failed....: 0.00%   0 out of 1503
```

```
✓ closet 200
✓ recommend 200
✓ weather 200
```

```
HTTP
http_req_duration.....: avg=685.39ms min=186.93ms med=202.76ms max=3.34s p(90)=3.19s p(99)=3.22s
  { endpoint:closet_all }.....: avg=199ms   min=186.93ms med=194.86ms max=363.87ms p(90)=213.63ms p(99)=315.95ms
  { endpoint:outfits_recommend }: avg=304.23ms min=275.38ms med=291.67ms max=820.41ms p(90)=315.95ms p(99)=315.95ms
  { endpoint:weather }.....: avg=2.52s   min=196.96ms med=3.19s   max=3.34s   p(90)=3.21s   p(99)=3.21s
  { expected_response:true }...: avg=685.39ms min=186.93ms med=202.76ms max=3.34s p(90)=3.19s p(99)=3.22s
http_req_failed.....: 0.00% 0 out of 1504
http_reqs.....: 1504    4.961839/s
```

```
EXECUTION
iteration_duration.....: avg=878.76ms min=187.56ms med=292.35ms max=3.94s p(90)=3.19s p(99)=3.22s
iterations.....: 1503    4.95854/s
vus.....: 1      min=1      max=8
vus_max...: 22     min=22     max=22
```

```
NETWORK
data_received.....: 16 MB 54 kB/s
data_sent.....: 963 kB 3.2 kB/s
```

```
running (5m03.1s), 00/22 VUs, 1503 complete and 0 interrupted iterations
closet  ✓ [=====] 00/12 VUs 5m0s 3.00 iters/s
recommend  ✓ [=====] 00/05 VUs 5m0s 1.00 iters/s
weather  ✓ [=====] 00/05 VUs 5m0s 1.00 iters/s
PS C:\Users\alexa\Documents\GitHub\Weather-to-Wear-app-backend>
```

Non-functional Requirements Testing

02 Scalability

Burst Upload

```
execution: local
script: burst-uploads.k6.js
output: -  
  
scenarios: (100.00%) 1 scenario, 12 max VUs, 5m30s max duration (incl. graceful stop):
* uploads: 0.10 iterations/s for 5m0s (maxVUs: 6-12, exec: uploadOne, gracefulStop: 30s)  
  
THRESHOLDS  
  
e2e_total_ms
✓ 'p(50)<9000' p(50)=8110.5
✓ 'p(95)<12000' p(95)=8559.6  
  
http_req_failed
✓ 'rate<0.02' rate=0.00%  
  
pipeline_total_ms
✓ 'p(50)<7000' p(50)=1027
✓ 'p(95)<10000' p(95)=1144.65  
  
TOTAL RESULTS  
  
checks_total.....: 60      0.192906/s
checks_succeeded.: 100.00% 60 out of 60
checks_failed....: 0.00%   0 out of 60  
  
✓ upload 2xx
✓ upload returned id  
  
CUSTOM
e2e_total_ms.....: avg=8150.966667 min=7764    med=8110.5   max=9194   p(90)=8419.6   p(95)=8559.6
pipeline_total_ms.: avg=959.466667 min=206       med=1027     max=1202   p(90)=1131.1   p(95)=1200
upload_latency_ms.: avg=7191.233333 min=6939     med=7060.5   max=8370   p(90)=7690.7   p(95)=8559.6  
  
HTTP
http_req_duration.: avg=2.51s      min=190.7ms   med=410.72ms  max=7.68s   p(90)=7.06s   p(95)=7.68s
  { expected_response:true }.: avg=2.51s      min=190.7ms   med=410.72ms  max=7.68s   p(90)=7.06s   p(95)=7.68s
http_req_failed....: 0.00%   0 out of 92
http_reqs.....: 92      0.295789/s  
  
EXECUTION
iteration_duration.: avg=8.57s      min=8.04s   med=8.54s   max=9.65s   p(90)=8.79s   p(95)=9.65s
iterations.....: 30      0.096453/s
vus.....: 0      min=0      max=1
vus_max.....: 6      min=6      max=6  
  
NETWORK
data_received.....: 758 kB 2.4 kB/s
data_sent.....: 215 kB 691 B/s  
  
Running (5m11.0s), 00/06 VUs, 30 complete and 0 interrupted iterations
uploads ✓ [=====] 00/06 VUs 5m0s 0.10 iters/s
S C:\Users\kylea\Documents\w2w\Weather-to-Wear\app-backend>
```

Non-functional Requirements

Testing

02 Scalability

SLOs

- ≥ 20 concurrent users with ≥ 5 req/s within latency SLOs
- Bursts of 6 uploads/min for 5 minutes with each completing in 10 seconds.

Non-functional Requirements Testing

03 Security

PASS tests/security/password-security.security.test.ts (7.361 s)

Security: Password Hashing Tests

Password Hashing Implementation

- ✓ should hash passwords using bcrypt (179 ms)
- ✓ should use appropriate salt rounds (minimum 10) (170 ms)
- ✓ should generate different hashes for the same password (341 ms)
- ✓ should verify hashed passwords correctly (487 ms)
- ✓ should handle empty password gracefully (659 ms)
- ✓ should handle special characters in passwords (343 ms)
- ✓ should handle unicode characters in passwords (339 ms)
- ✓ should be case-sensitive (454 ms)

Password Validation Rules

- ✓ should enforce minimum length requirement (23 ms)
- ✓ should require uppercase letter (9 ms)
- ✓ should require lowercase letter (8 ms)
- ✓ should require special character (9 ms)
- ✓ should accept valid strong password (269 ms)

Password Storage Security

- ✓ should never store plaintext passwords (170 ms)
- ✓ should use salt for password hashing (342 ms)

Timing Attack Prevention

- ✓ should take similar time for password comparison regardless of

Authentication Flow Security

- ✓ should not reveal whether email exists during login attempts (4 ms)
- ✓ should handle password comparison errors gracefully (10 ms)

Hash Format Validation

- ✓ should handle invalid hash formats during comparison (2 ms)
- ✓ should handle corrupted hash gracefully (172 ms)
- ✓ should validate bcrypt hash format structure (170 ms)

Edge Cases

- ✓ should handle very long passwords (336 ms)
- ✓ should handle password with only minimum requirements (347 ms)

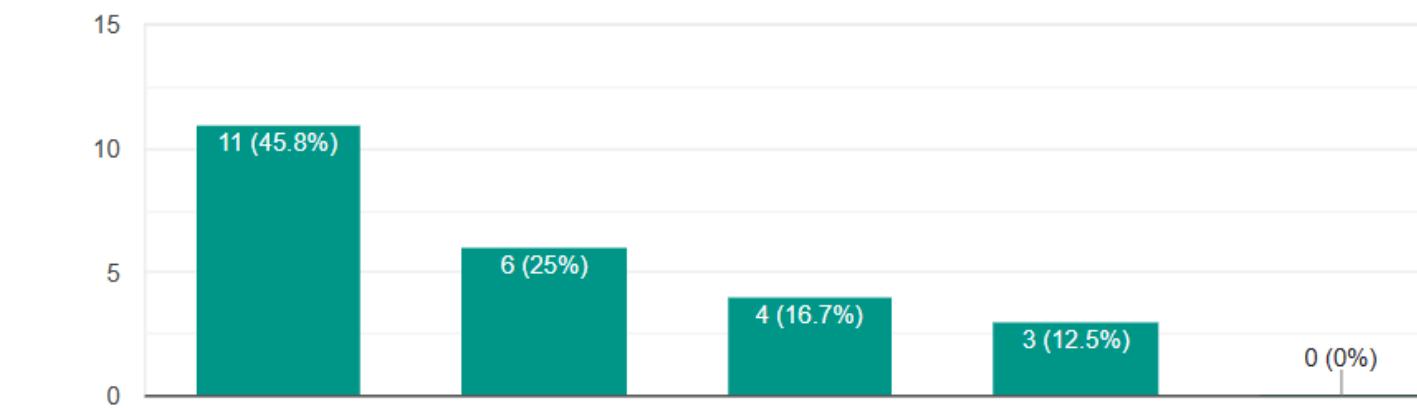
Non-functional Requirements Testing

04 Usability

How easy was it to learn how to use Weather To Wear? (1 being easiest)

[Copy chart](#)

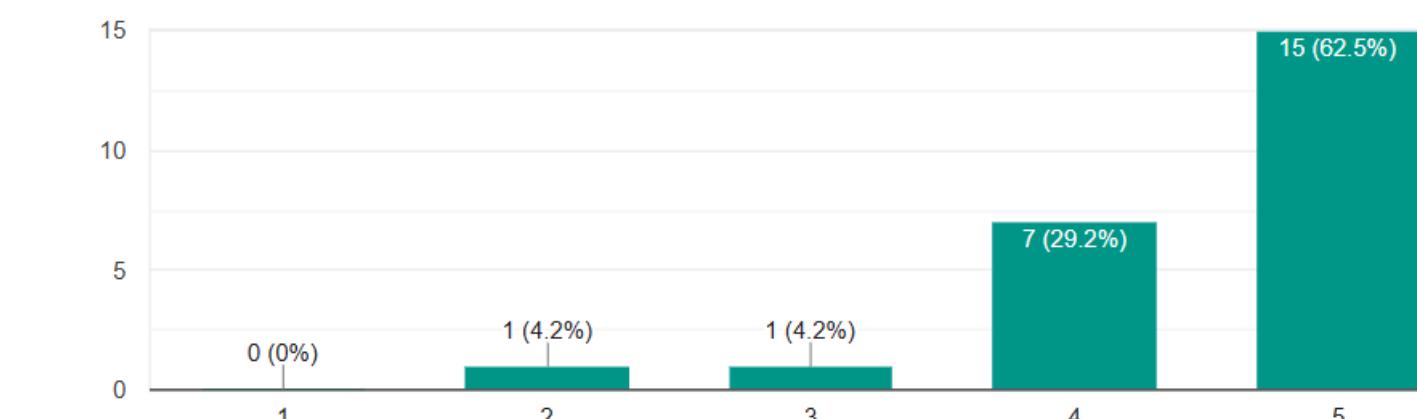
24 responses



Overall, how satisfied are you with the app?

[Copy chart](#)

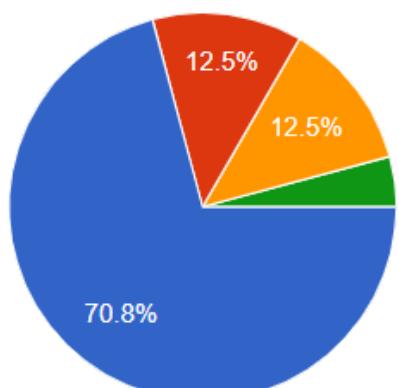
24 responses



I would continue to use this app in the future.

[Copy chart](#)

24 responses

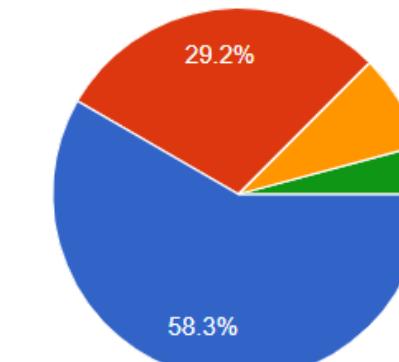


- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

The layout of the app is clear and intuitive.

[Copy chart](#)

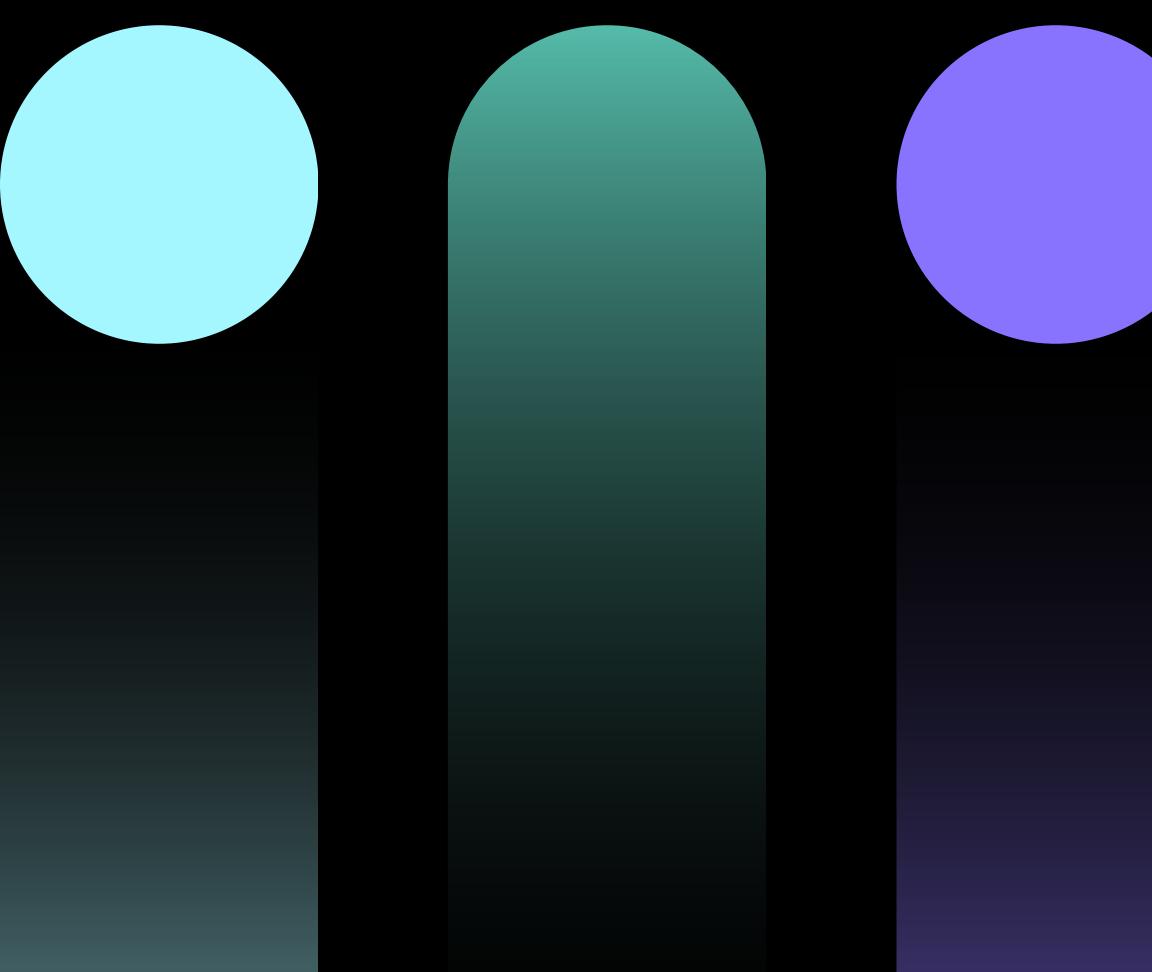
24 responses



- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

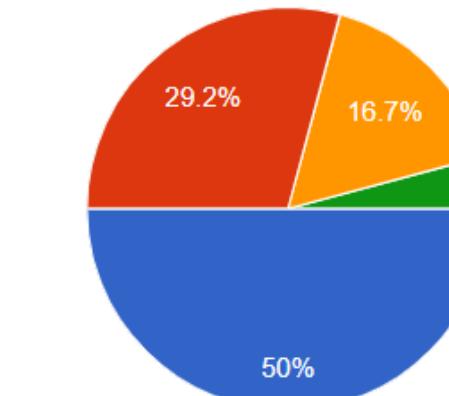
Non-functional Requirements Testing

04 Usability



I was able to find what I was looking for without difficulty.

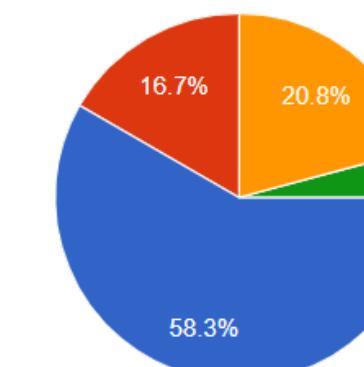
24 responses



- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

The outfit recommendations were useful and relevant.

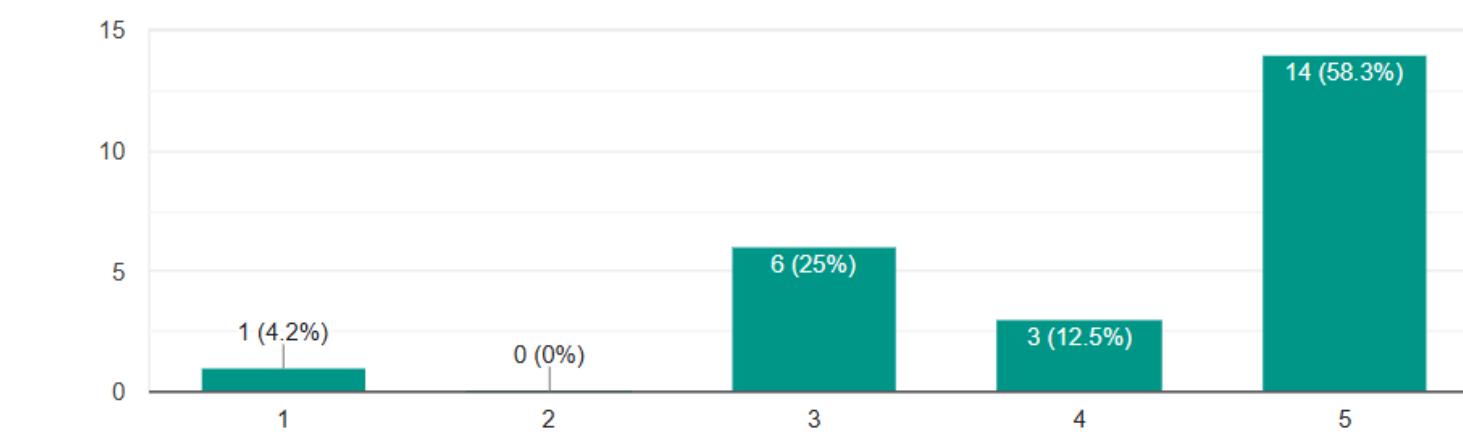
24 responses



- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

How satisfied are you with the speed and performance of the app?

24 responses



01 Wow Factors

Try On Features

01.1 Try On Avatar

- Users can see how an outfit looks like on a 2D mannequin
- Drag, Scale, and Rotate options allow the user to ensure the clothing items are fitted well onto the mannequin
- The coordinates for each item are saved to prevent constant readjustments

01.1 Try On Yourself

- Using an external API, ‘FashionAI’ we are able to allow users to have an accurate look into what the outfit will look like on their body

02 Wow Factors

Collaborative Style Intelligence

- Inspiration outfit recommendations are generated based off what a user likes off of other people's social media posts.
- The recommendation system then analyzes clothing item tags from these liked items and uses a weighted warmth algorithm to assemble an outfit based on what weather it should be worn in, style, and overall colour scheme.
- Only outfits that fit the selected temperature range are returned, ensuring recommendations are both personalized and seasonally appropriate.

03 Wow Factors

Fashion Time Machine

- This feature allows users to not only have an outfit ready per day, but lets them plan ahead for any occasion!
- With predictive wardrobe planning for upcoming trips and events with precise weather forecasts using our weather API, 'OpenMeteo'
- Calendar showing complete outfit recommendations for upcoming events and trips with adaptive modifications as forecasts update
- Pack for a trip feature, which allows users to select items and outfits and suggests missing items based on the predicted weather

04 Wow Factors

Image Processing

04.1 Background Remover

- Makes use of u2net, which is a predefined model for background removal. A python script was written to normalize and resize each image, process it, then composite it with original RGB to therefore produce and RGBA cut-out (jpeg).

04.2 Colour Extraction

- Once the background on the image is removed, it is downscaled in size to reduce noise from all the tiny details. K-means is then used to cluster the pixels and pick the 3 most dominant colors in the clothing item.

04 Wow Factors

Image Processing

04.3 Multitasking for batch uploads

- A worker queue was deployed using BullMQ on Redis. Scalable workers were used to concurrently tackle images uploaded on the queue, where they call the U2net background removal and from there the color extracting script. Any retries are configured using exponential backoff, timeouts, and idempotent writes.

Deployment

Understand our system

Frontend: Client loads React PWA via CloudFront; origin is private S3 (accessed through OAC).

Backend: App Runner hosts the REST API; image pulled from ECR; secrets from Secrets Manager.

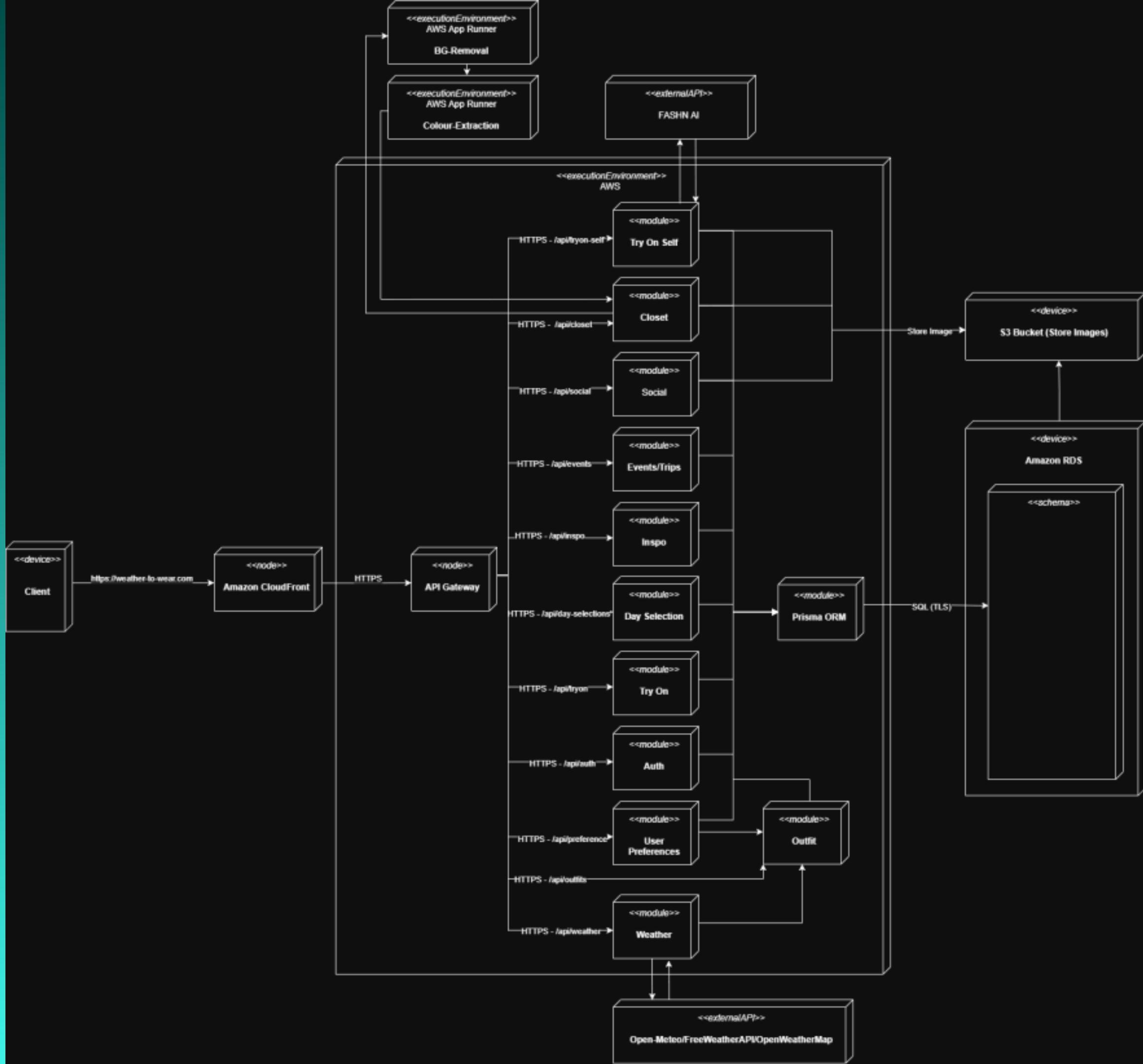
Data path: Backend connects privately to RDS PostgreSQL via App Runner VPC Connector inside the VPC (no public DB).

Image pipeline: Backend calls bg-removal and color-extract microservices (App Runner) via simple HTTP endpoints.

Media: Processed images stored in private S3 Uploads; users fetch via CloudFront (Uploads CDN).

Security: TLS everywhere, least-privilege IAM, private buckets with OAC, private RDS.

Outcome: Fast delivery (CDN), secure private data paths, and independent scaling of API and image services.



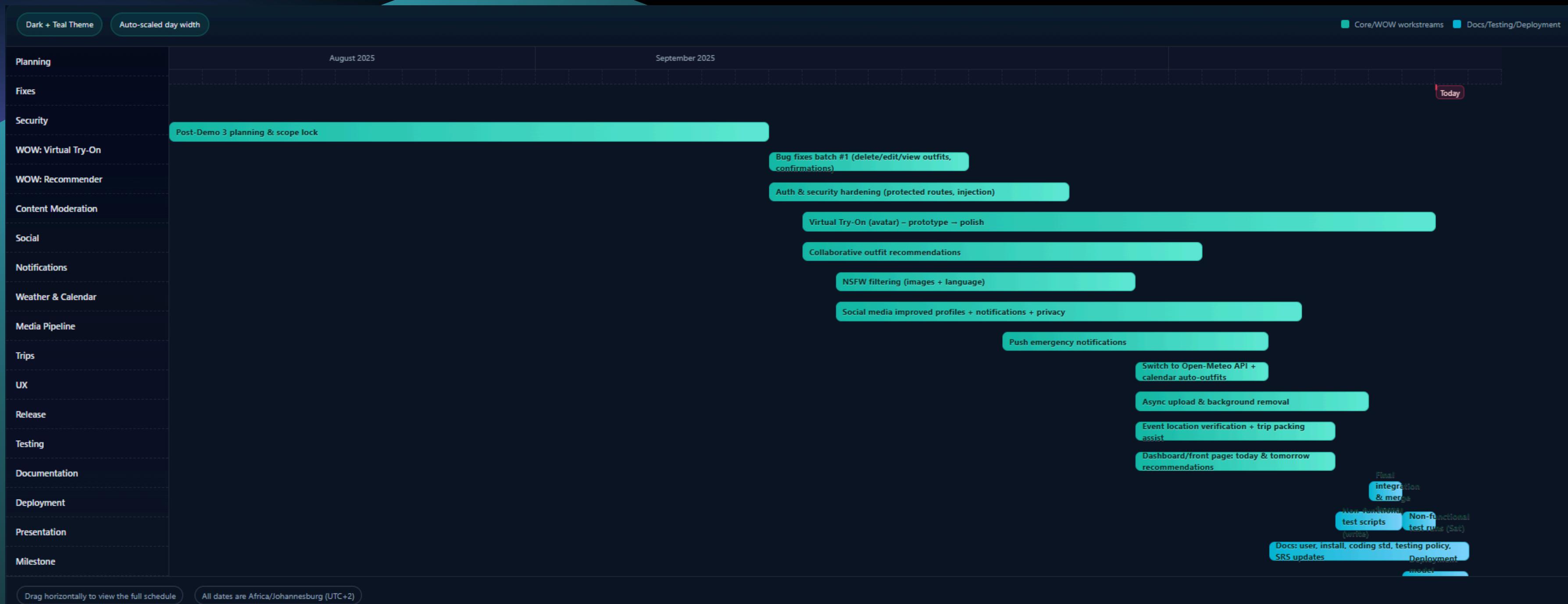
Project Management

Burn Down Chart



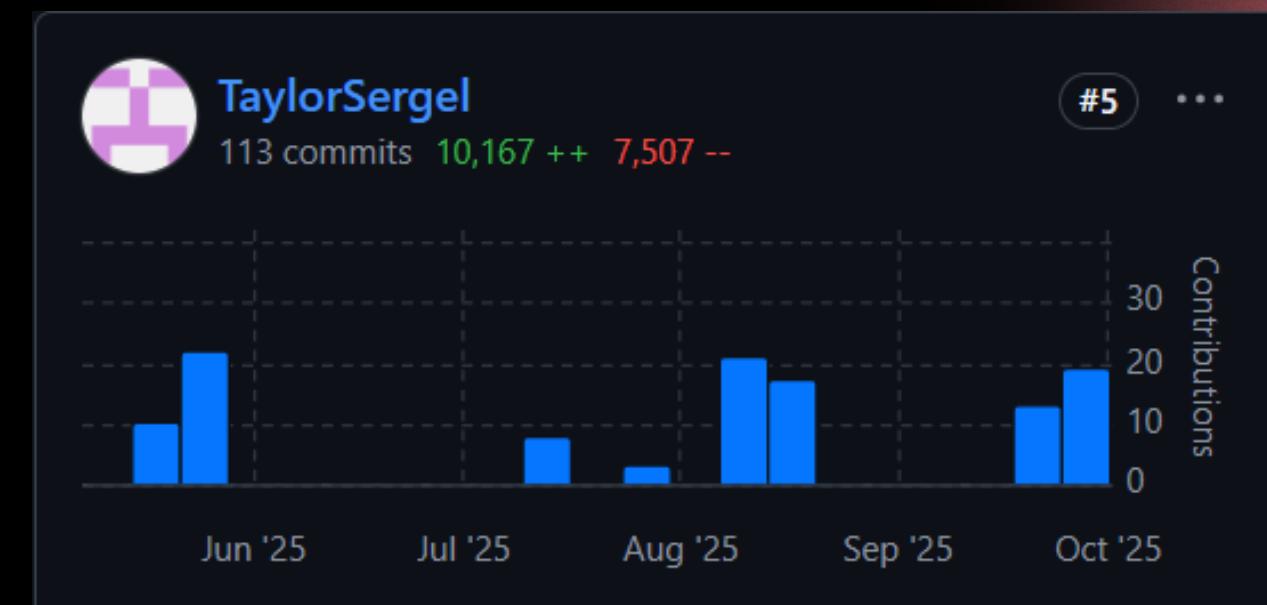
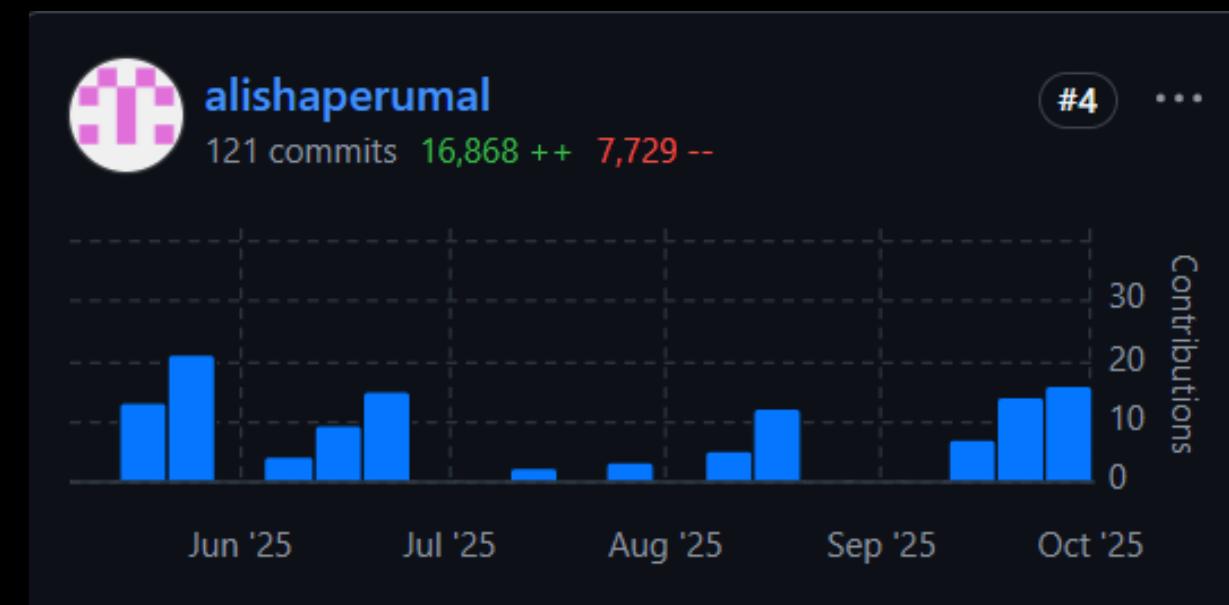
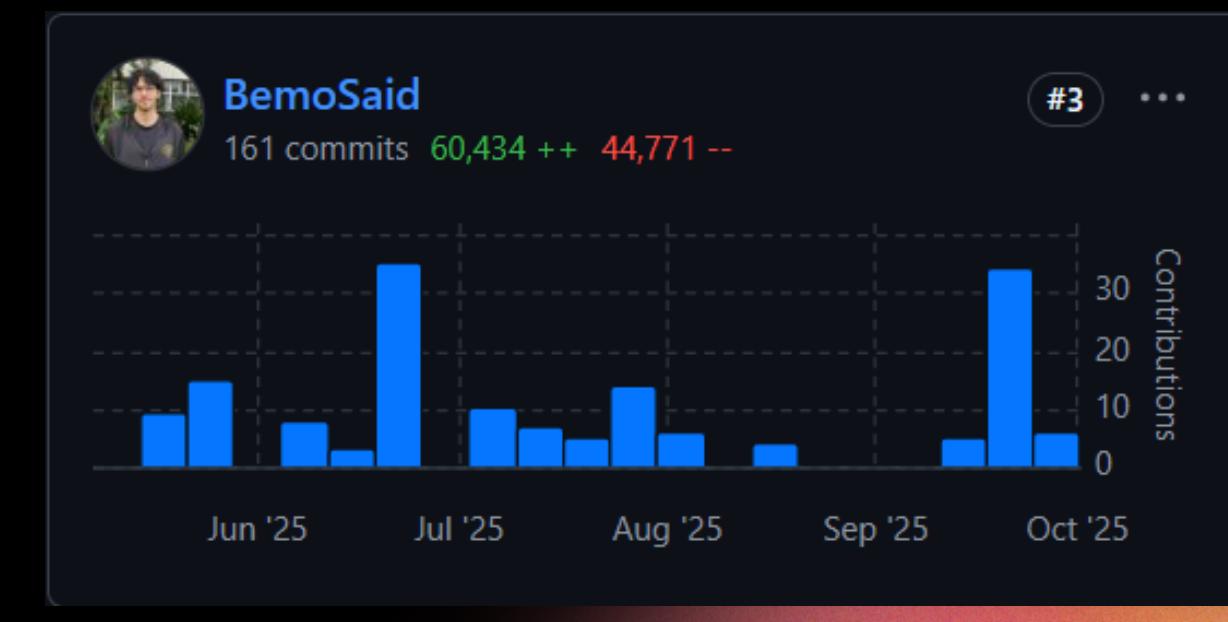
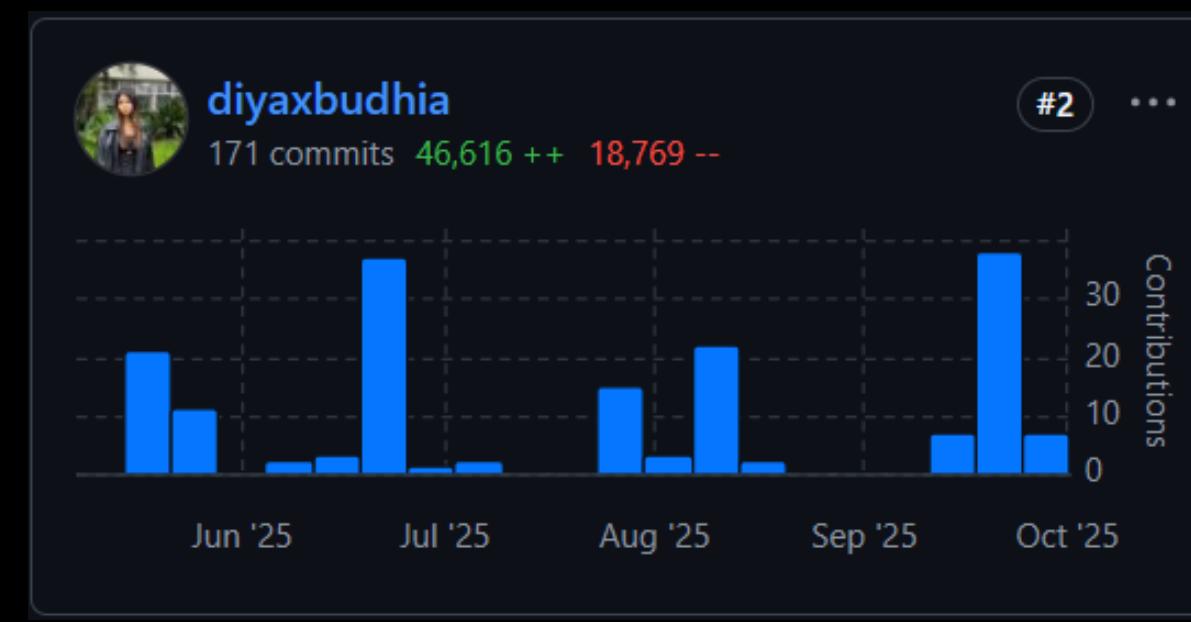
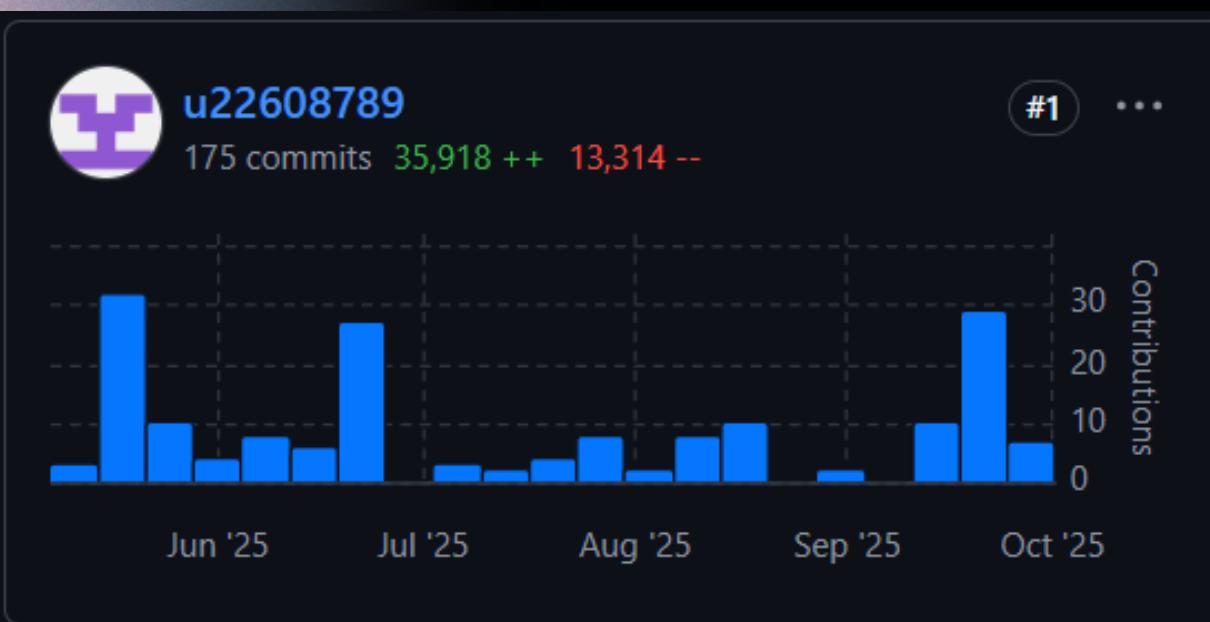
Project Management

Gantt Chart



Project Management

Git Contributions and SCRUM frequency



01 Testing

Unit Tests

File	% Stmt	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	77.47	54.63	83.4	80.22	
src	100	100	100	100	
prisma.ts	100	100	100	100	
src/middleware	50	23.52	0	50	
upload.middleware.ts	50	23.52	0	50	20-23, 34-44
src/modules/auth	89.55	83.33	100	88.88	
auth.middleware.ts	77.41	75	100	77.41	31-32, 39-40, 46-51
auth.service.ts	100	100	100	100	
auth.utils.ts	100	100	100	100	
src/modules/closet	77.88	52.38	90	78.01	
closet.controller.ts	72.17	43.75	90.9	71.92	...05-106, 127-128, 163, 170-171, 205-206, 226-227, 235, 237, 239, 241, 243, 245, 262, 271-285
closet.route.ts	100	100	100	100	
closet.service.ts	82.85	61.29	88.88	84.12	46, 51, 75, 90-91, 112-113, 194-199
src/modules/events	91.33	86.48	100	94.21	
events.controller.ts	91.33	86.48	100	94.21	32, 70, 133, 147-148, 216, 249
src/modules/outfit	79.66	60.42	80.86	82.85	
collabFiltering.ts	58.06	56.52	44.44	62.22	51-57, 72-87
itemItemKnn.ts	97.91	50	100	100	17-34, 40-80
outfit.controller.ts	82.75	60.97	80	82.85	21-30, 72, 84, 97, 119, 135, 181, 240, 247-260
outfit.service.ts	85.24	92.85	91.66	84.61	153, 193, 217, 234-235, 239-241
outfitRecommender.service.ts	78.91	56.72	80.3	83.6	28-33, 137-138, 148-149, 245-247, 254, 256, 319-338, 356-379, 396-398
src/modules/packing	100	89.74	100	100	
packing.controller.ts	100	89.74	100	100	57, 85, 129, 153
src/modules/social	65.35	34.78	75	68.35	
social.controller.ts	65.35	34.78	75	68.35	...36, 156-158, 168-184, 216-225, 270, 297-301, 326, 354, 438-443, 450-455, 460-465, 470-496
src/modules/userPreference	100	100	100	100	
userPref.controller.ts	100	100	100	100	
src/modules/weather	74.09	41.53	95	79.22	
weather.service.ts	74.09	41.53	95	79.22	...36-153, 239, 288, 336, 379, 434, 467, 510-518, 537, 587, 619-620, 634, 649-666, 677, 708-709
src/utils	43.18	14.28	25	47.5	
logger.ts	100	100	100	100	
s3.ts	39.02	14.28	25	43.24	20-31, 35, 62-96

```
Test Suites: 13 passed, 13 total
Tests:      206 passed, 206 total
Snapshots:  0 total
Time:       19.301 s
Ran all test suites.
```

02 Testing

Integration Tests

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	62.43	39.17	66.14	65.52	
src	98.03	50	50	98.03	
app.ts	97.91	50	50	97.91	54
prisma.ts	100	100	100	100	
src/middleware	85.71	76.72	80	86.66	
nsfw.middleware.ts	90.19	80.8	100	91.66	184-188,205-208
upload.middleware.ts	66.66	52.94	33.33	66.66	12-13,20-23,39-41
src/modules/auth	91.07	76.92	92.85	90.47	
auth.controller.ts	94.11	87.5	100	93.54	62-63
auth.middleware.ts	77.41	66.66	100	77.41	31-32,39-40,46-51
auth.routes.ts	90.9	100	0	90.9	25
auth.service.ts	100	100	100	100	
auth.utils.ts	100	50	100	100	5
src/modules/closet	71.85	36.5	95	74.86	
closet.controller.ts	64.34	31.25	90.9	64.91	...63,170-171,192,285-206,213,226-227,233,239,243,245,262,275-276,285
closet.route.ts	100	100	100	100	
closet.service.ts	78.57	41.93	100	87.3	48-51,75,90-91,112-113
src/modules/daySelection	90.12	75	100	97.22	
daySelection.controller.ts	86.11	62.5	100	100	9,29,38-48
daySelection.routes.ts	88.88	100	100	92	26,26
daySelection.service.ts	100	100	100	100	
src/modules/events	71.53	45.94	100	72.51	
events.controller.ts	69.29	45.94	100	70.24	...48,158-159,179-180,189-190,196-197,216,224-225,230-231,239-240,249
events.route.ts	100	100	100	100	
src/modules/inspo	59.61	40.09	62.36	61.05	
inspo.controller.ts	74.13	33.33	87.5	75.43	42-43,65-66,84-93,106-107,122-123,136-137
inspo.routes.ts	100	100	100	100	
inspo.service.ts	58.12	48.9	52.63	58.33	68-111,195-197,199-201,203-205,207-209,219,313-316,340,353-441
inspoRecommender.service.ts	57.02	36.24	65.95	58.82	...71,495,501,507,513,571-572,576-577,615,623,629-631,649,655-656,727
src/modules/outfit	70.56	48.76	72.17	75.95	
collabFiltering.ts	66.12	47.82	44.44	71.11	54-55,72-87
itemItemKnn.ts	12.5	0	0	15.78	17-27,32-46,51-54,65-80
outfit.controller.ts	61.2	43.9	86.66	66.66	...21,130-131,137,181,193-194,198-199,208,219-220,225,233-242,247-260
outfit.routes.ts	100	100	100	100	
outfit.service.ts	73.77	60.71	83.33	86.53	54,128-135,239-241
outfitRecommender.service.ts	81.78	53.8	84.84	86.47	28-33,137-138,149,247,254-259,336-338,356-379,396-398
src/modules/packing	79.02	45.2	100	82.64	
packing.controller.ts	72.97	51.28	100	71.42	19-20,40-41,67-68,73-74,84-85,94-95,110-111,114-115,138-139,144-145
packing.routes.ts	100	100	100	100	
packing.service.ts	83.33	38.23	100	97.61	57
src/modules/social	64.42	36.36	66.17	66.66	
social.controller.ts	60.23	34.78	79.16	63.71	...12,326,352,366-368,413,424,433,438-443,450-455,460-465,473-474,496
social.route.ts	77.04	12.5	80	77.04	28-22,32-34,40-41,124-131
social.service.ts	66.43	39.77	52.94	67.18	55-56,213-214,229,235,241,257-293,361,366,371-375,385-389,447,490-597
src/modules/tryon	35.89	0	0	37.83	
tryon.controller.ts	13.63	0	0	14.28	7-17,21-36
tryon.routes.ts	100	100	100	100	
tryon.service.ts	40	0	0	44.44	6-9,20-41
src/modules/tryon-self	12.93	2.89	0	15.67	
tryon-self.controller.ts	3.94	0	0	5.17	14-186
tryon-self.routes.ts	100	100	100	100	
tryon-self.service.ts	11.11	3.96	0	13.25	28-130,145-167,171-176,180,184-307,312-315,323-332,341-352
src/modules/userPreference	75.6	70	100	74.35	
userPref.controller.ts	70.58	70	100	68.75	14-15,33-34,44-45,51-52,72-73
userPref.routes.ts	100	100	100	100	
src/modules/users	68.49	46.66	75	71.42	
users.controller.ts	62.06	46.66	88	63.41	11-12,30,36-37,65,71-83
users.routes.ts	100	100	100	100	
users.service.ts	83.33	100	66.66	83.33	15
src/modules/weather	50.24	27.31	59.09	53.73	
weather.controller.ts	48.88	42.85	58	48.88	30-31,42-57,64-74
weather.routes.ts	100	100	100	100	
weather.service.ts	49.3	25.64	60	53.24	...39,587,602-605,611-614,619-620,627-666,677,692-695,700-703,708-709
src/utils	39.28	14.28	0	39.28	
logger.ts	100	100	100	100	
sightengine.ts	32	14.28	0	32	11-15,23-34,38-49

Any Questions

Have any questions about Weather to Wear?

We'd love to answer them and share more about our project.