

Day 3: API Integration and Data Migration

-Presented by Rida Fatima.

Agenda

Topics Covered 

Objectives:

Successfully integrated API data into Sanity CMS.

Completed data migration from Template 6 API.

Ensured a fully functional CMS backend for managing Nike Store content.

Set up Sanity CMS:
Created and connected the Sanity CMS project.

1

Securely added the API token:
Generated and stored the API token using environment variables.

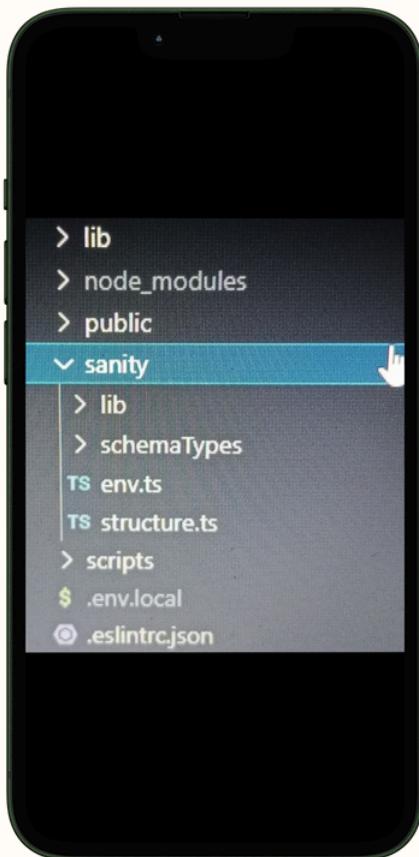
2

Implemented provided schema
Designed schemas for:
Products (e.g., shoes, shorts, clothing).
Categories and inventory.

3

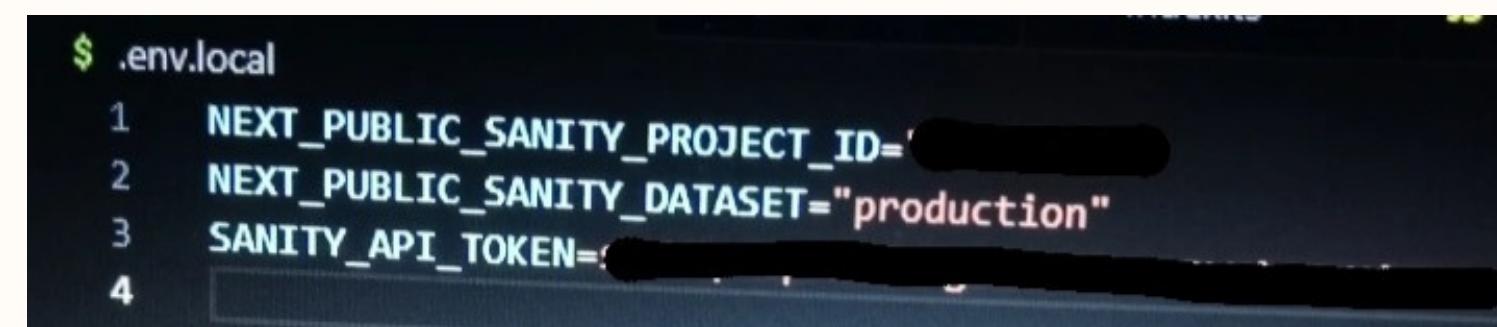
1. Set up Sanity Cms

- Created a new Sanity CMS project on sanity.io and integrated it with the local Next.js project.
- Used the Sanity CLI to initialize the project locally, automatically generating the required configuration files in a dedicated Sanity folder.



2. Secure API Token Integration :

- Generated an API token from the Sanity dashboard to enable secure access to the CMS.
- Stored the project ID, dataset name, and API token in the .env.local file to maintain security and prevent unauthorized access.



```
$ .env.local
1 NEXT_PUBLIC_SANITY_PROJECT_ID=[REDACTED]
2 NEXT_PUBLIC_SANITY_DATASET="production"
3 SANITY_API_TOKEN=[REDACTED]
4 [REDACTED]
```

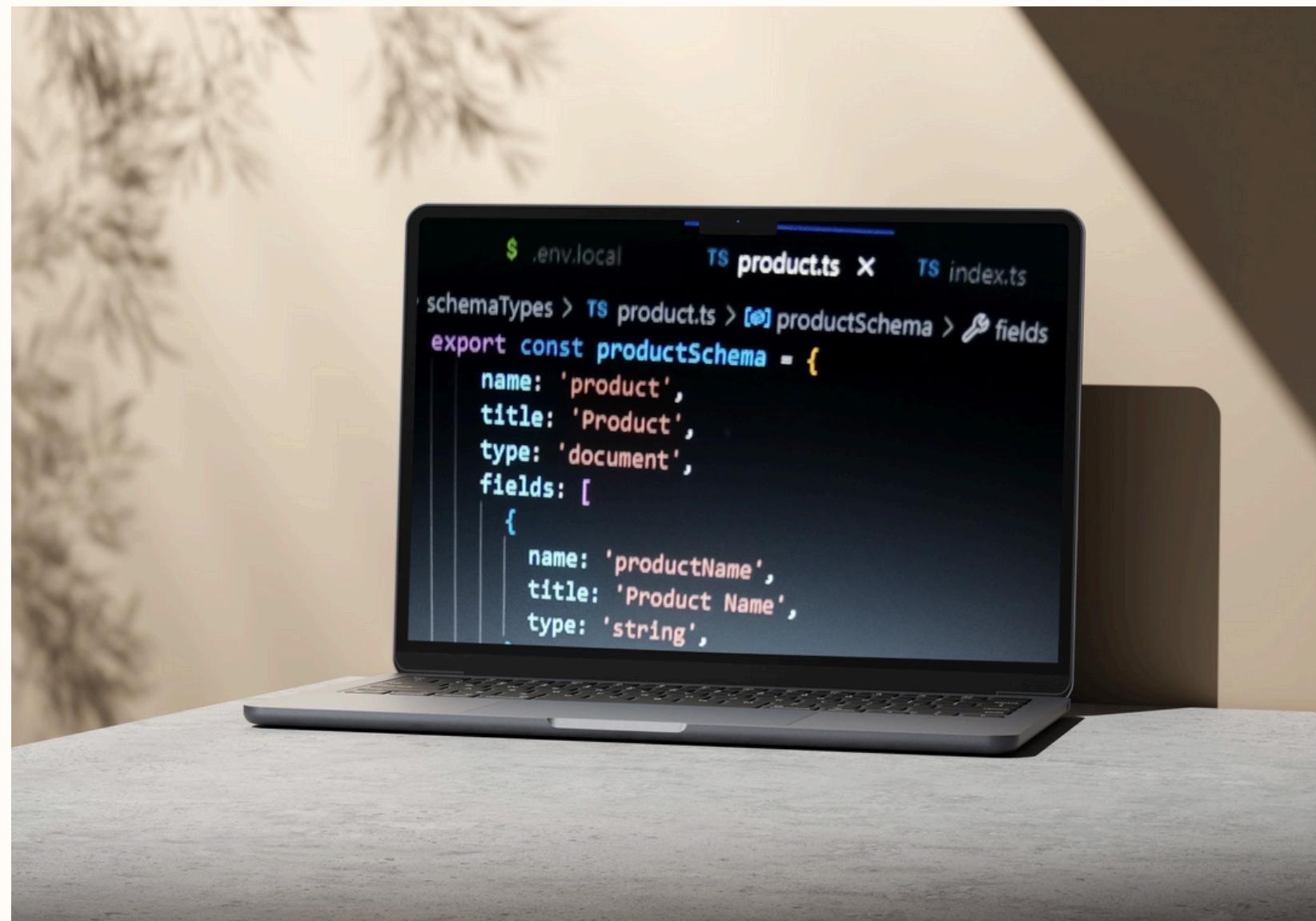
A screenshot of a terminal window displaying the contents of a .env.local file. The file contains environment variables: NEXT_PUBLIC_SANITY_PROJECT_ID, NEXT_PUBLIC_SANITY_DATASET set to "production", and SANITY_API_TOKEN. All three variable values are completely redacted with black boxes. The terminal prompt '\$' is visible at the top left.

3. Creating the Schema:

- Imported the provided product.ts schema from the specified source link into the Sanity project.
- Placed the schema file in the schemaTypes folder, which is responsible for defining the structure and fields for the product data in the CMS.
- Ensured the schema was properly configured to include necessary fields such as product name, description, price, category, and image URLs.
- Validated the schema within the Sanity Studio to confirm it was functional and compatible with the CMS.

• SchemaTypes

/product.ts

A photograph of a dark laptop sitting on a light-colored wooden desk. The laptop screen displays a code editor with a dark theme, showing a portion of a TypeScript file named 'product.ts'. The code defines a schema type 'productSchema' with properties like 'name', 'title', 'type', and 'fields'.

```
$ .env.local      TS product.ts X  TS index.ts
schemaTypes > TS product.ts > [e] productSchema > fields
export const productSchema = {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'productName',
      title: 'Product Name',
      type: 'string',
    }
  ]
}
```

4. Data Migration Script :

- Developed a script named importSanityData.mjs and placed it in the scripts folder of the project.
- The script was designed to automate the migration process by fetching data from the provided API.
- Imported the retrieved data directly into Sanity CMS, ensuring it aligned with the predefined schema.
- Tested the script to confirm successful data transfer and resolved any mapping issues during the migration process.

• Scripts

/importSanityData.mjs

A photograph of a silver laptop resting on a light-colored wooden surface. The screen displays a terminal window with code. The tab bar at the top of the terminal shows tabs for '.env.local', 'TS products', 'TS index.ts', and 'JS importSanityData.mjs', with 'JS importSanityData.mjs' being the active tab. The code in the terminal is as follows:

```
scripts > JS importSanityData.mjs > uploadImageToSanity > asset
22  async function uploadImageToSanity(imageUrl) {
23    const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
24    const buffer = Buffer.from(response.data);
25    const asset = await client.assets.upload('image', buffer, [
26      filename: imageUrl.split('/').pop()
27    ]);
28    console.log(`Image uploaded successfully: ${asset._id}`);
29    return asset._id;
30  } catch (error) {
31    console.error(`Failed to upload image: ${imageUrl}, ${error}`);
32    return null;
33  }
34
35
36
37
38  async function importData() {
39    try {
40      console.log('migrating data please wait...');
41    } catch (error) {
42      console.error(`Failed to migrate data: ${error}`);
43    }
44  }
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
89
90
91
92
93
94
95
96
97
98
99
99
100
101
102
103
104
105
106
107
108
109
109
110
111
112
113
114
115
116
117
118
119
119
120
121
122
123
124
125
126
127
128
129
129
130
131
132
133
134
135
136
137
138
139
139
140
141
142
143
144
145
146
147
148
149
149
150
151
152
153
154
155
156
157
158
159
159
160
161
162
163
164
165
166
167
168
169
169
170
171
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
188
189
189
190
191
192
193
194
195
196
197
198
199
199
200
201
202
203
204
205
206
207
208
209
209
210
211
212
213
214
215
215
216
217
218
219
219
220
221
222
223
223
224
225
226
227
227
228
229
229
230
231
232
233
233
234
235
235
236
236
237
237
238
238
239
239
240
240
241
241
242
242
243
243
244
244
245
245
246
246
247
247
248
248
249
249
250
250
251
251
252
252
253
253
254
254
255
255
256
256
257
257
258
258
259
259
260
260
261
261
262
262
263
263
264
264
265
265
266
266
267
267
268
268
269
269
270
270
271
271
272
272
273
273
274
274
275
275
276
276
277
277
278
278
279
279
280
280
281
281
282
282
283
283
284
284
285
285
286
286
287
287
288
288
289
289
290
290
291
291
292
292
293
293
294
294
295
295
296
296
297
297
298
298
299
299
300
300
301
301
302
302
303
303
304
304
305
305
306
306
307
307
308
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
```

5. Running the Migration Script :

- Added a new script named "import-data" to the package.json file to simplify the execution of the migration process.
- The script was named "import-data", allowing easy execution through the command `npm run import-data`.
- Ran the script to initiate the migration, which fetched data from the API and imported it into Sanity CMS, ensuring everything was correctly populated.



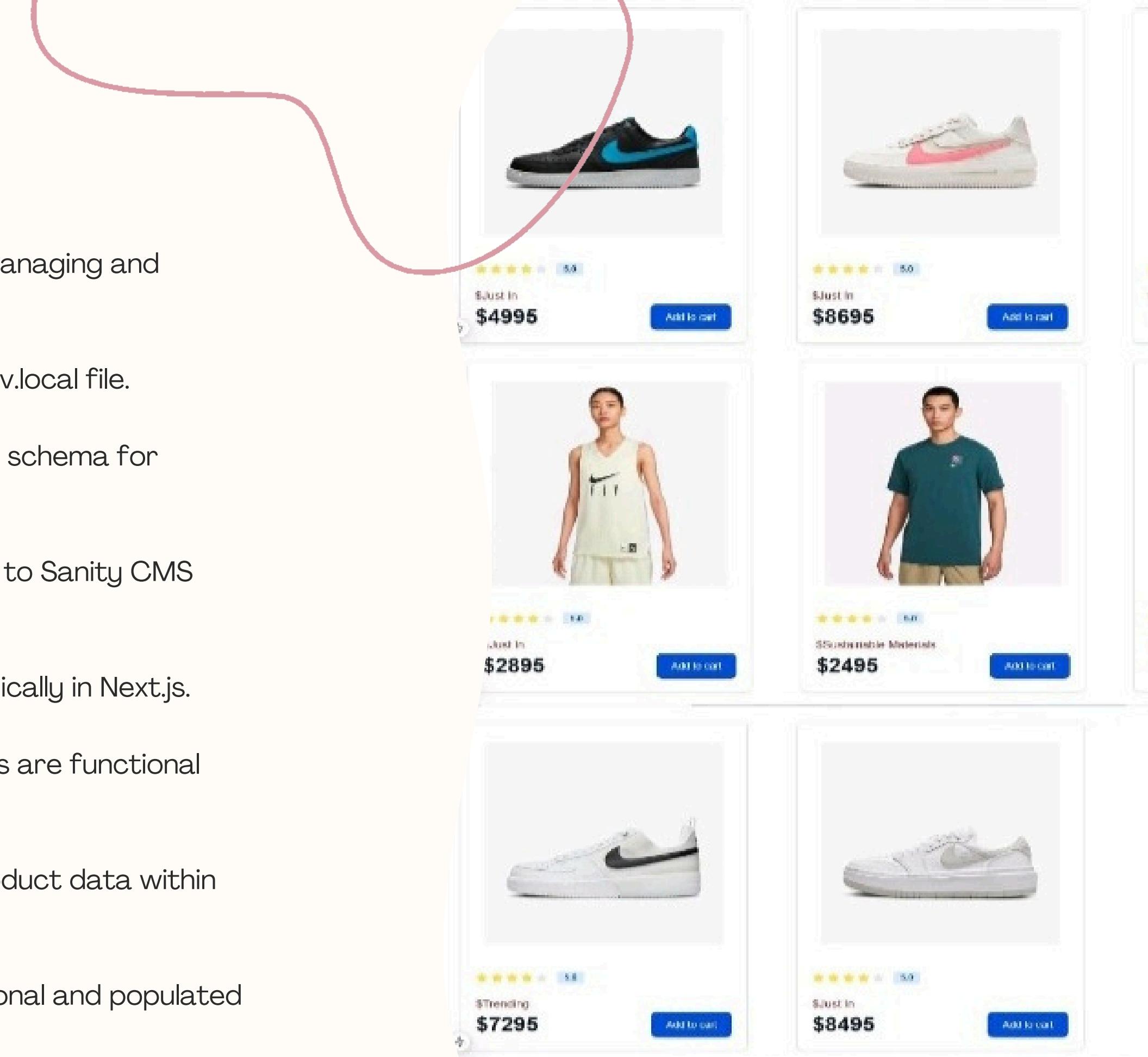
"import-data" : "node scripts/importSanityData.mjs"

Self Validation Checklist ✓

Sanity CMS Setup	API Token Integration	Schema Creation	Data Migration Script	Final Check ✓
✓	✓	✓	✓	✓✓

• Outcomes :

1. Sanity CMS Integration: Linked Sanity CMS with Next.js for managing and displaying content.
2. API Token Setup: Securely configured the API token in the .env.local file.
3. Schema Creation: Implemented and validated the product.ts schema for product data.
4. Data Migration: Automated the data migration from the API to Sanity CMS using a migration script.
5. API Integration: Integrated API to fetch product data dynamically in Next.js.
6. Final Submission: Completed the project, ensuring all features are functional and ready for submission.
7. Data Management: Optimized the process for managing product data within the CMS.
8. Fully Functional Backend: Ensured Sanity CMS is fully operational and populated with accurate data.



Project Name : Nike Store
Presented by Rida Fatima
Hackathon 3 Day 3 Task 3
Thursday Class: 9am to 12pm

Respectful Sir: Ameen Alam, Sir Hamza Alvi, and Sir Fahad Khan, I appreciate your guidance and support.

Thank you!

Looking forward to the next steps in this project!