

PromoPilot – Campaign Performance Management System for retail chains

1. Introduction:

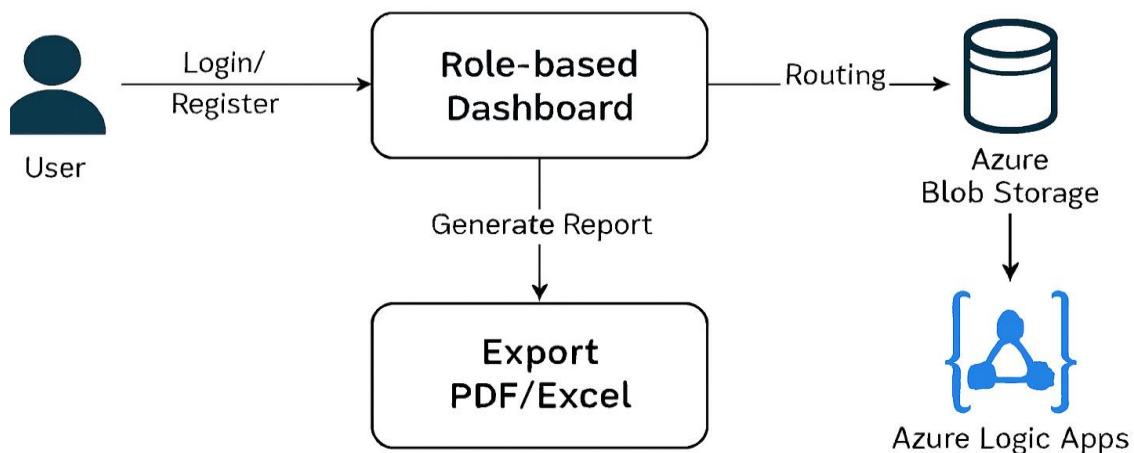
PromoPilot is a web-based system for managing and analyzing offline promotional campaigns in retail stores. While promotions happen physically in-store, all planning, execution, and performance tracking are handled online by marketing, finance, and store managers through role-based dashboards.

2. Architecture Overview

Tech Stack:

- Angular (standalone modules)
- ASP.NET Core Web API
- Azure Blob Storage, ACS, Logic Apps
- SQL Server (via EF Core)

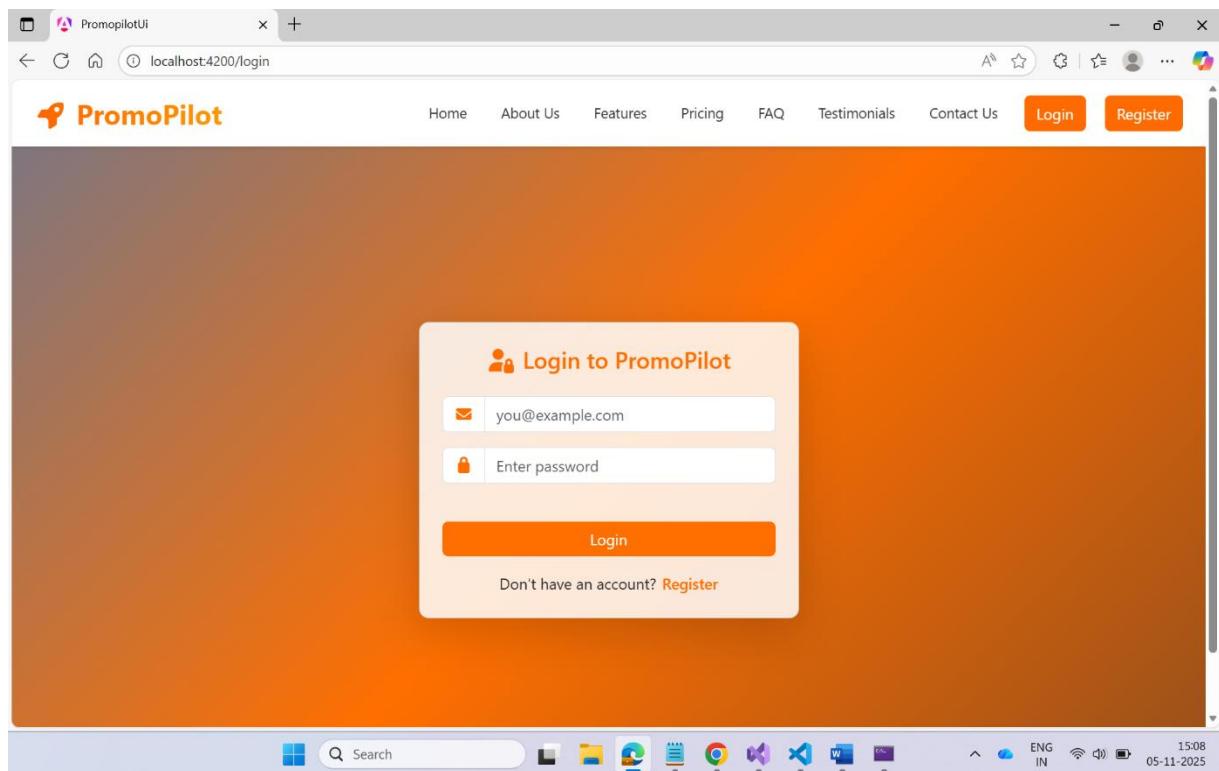
PromoPilot – System Architecture Diagram



3. Modules Implemented

| Module | Description |
|----------------------------|--|
| Login | Secure login with JWT, lockout, and role-based routing |
| Register | Role selection with ACS email confirmation |
| Home Page | Displays landing page with navigation links, feature highlights, and call-to-action buttons (Start Free Trial, Explore Features) |
| Campaign Planning | Create, update, schedule, cancel campaigns |
| Budget Allocation | Allocate and track spend per store with export |
| Engagement Tracking | Track coupon redemptions, product purchases during campaigns. |
| Execution Status | Update the campaigns progress at each store, including status and manager feedback. |

Login Module



Register Module

A screenshot of a web browser showing the registration page for PromoPilot. The title bar says "PromopilotUi" and the address bar shows "localhost:4200/register". The page features a dark blue header with the "PromoPilot" logo. Below the header is a large input form titled "Create your PromoPilot account". It includes fields for "Your username", "you@example.com", "Create password", and "Select Role". There is also an orange "Register" button and a link to "Already have an account? Login". The background of the page has a blurred gradient from brown to blue.

Home Page

A screenshot of a web browser showing the home page for PromoPilot. The title bar says "TippasamudramShaikMariyamun" and the address bar shows "localhost:4200". The page features a light gray header with the "PromoPilot" logo. Below the header is a main content area with a blue header "Welcome to PromoPilot" and the subtext "Track campaigns, engage customers, and grow your retail impact.". It includes two orange buttons: "Start Free Trial" and "Explore Features". The background of the main content area has a subtle pattern of orange dots. At the bottom of the page is a photograph of several people working together at a table, looking at laptops and tablets displaying data. The footer of the browser shows various icons and the date "07-11-2025".

TippasamudramShaikMariyamun... X PromopilotUi

localhost:4200

Key Features

Campaign Analytics
Real-time tracking and performance insights for marketing teams.

Finance Dashboard
Budget tracking, ROI analysis, and financial planning tools.

Store Manager Tools
Task checklists, inventory sync, and store-level campaign execution.

12:24 07-11-2025

TippasamudramShaikMariyamun... X PromopilotUi

localhost:4200

Pricing

Starter
Basic tools for small teams
₹999/month

Enterprise
Custom solutions for large retail chains
₹4999/month

Includes:

- ✓ Finance dashboard
- ✓ ROI tracking
- ✓ Multi-store support

[Start Trial](#)

12:24 07-11-2025

TippasamudramShaikMariyamun... X PromopilotUi

localhost:4200

FAQ

Who can use PromoPilot?

Is there a free trial?

Can PromoPilot integrate with our existing POS or CRM?

Yes, PromoPilot supports API-based integrations with major POS and CRM platforms.

Do you offer onboarding and training?

What Our Clients Say

Trusted by retail teams across India

12:25 07-11-2025

TippasamudramShaikMariyamun... X PromopilotUi

localhost:4200

What Our Clients Say

Trusted by retail teams across India



Anjali Mehta
Marketing Lead, TrendMart
"PromoPilot helped us align marketing and store ops like never before. Campaigns now launch in half the time."



Ravi Kumar
COO, StyleSpace
"The finance dashboard gave us real-time ROI clarity. We've optimized spend across 120+ stores."



Neha Kapoor
Retail Ops Head, UrbanRack
"Our store managers love the checklist tools. Execution is smoother and more consistent chain-wide."

12:25 07-11-2025

Contact Us

PromoPilot
Your campaign partner in retail excellence

Email: support@promopilot.io
Phone: +91 98765 43210
Location: Hyderabad, India
Business Hours: Mon–Fri, 9:00 AM to 6:00 PM IST

For demo requests, partnership inquiries, or technical support, reach out anytime. We typically respond within 24 hours.

Hyderabad Map showing various landmarks like Snow World, Sudarshan Theatre 35MM, and Mehfil.

Campaign Planning Module

PromoPilot

Marketing Dashboard

All Campaigns

| CampaignID | CampaignName | StartDate | EndDate | Products | Stores | Actions |
|------------|------------------|--------------|--------------|------------------------------|------------------|--|
| 31 | Sankranthi Offer | Oct 17, 2025 | Oct 20, 2025 | Rangoli, Groceries | Andhra Pradesh | View Edit Cancel |
| 34 | Diwali Offers | Oct 20, 2025 | Oct 26, 2025 | Crackers, Clothing | Hyderabad | View Edit Cancel |
| 6 | Diwali Dhamaka | Oct 20, 2025 | Nov 10, 2025 | Electronics, Home Appliances | Hyderabad, Bnglr | View Edit Cancel |

Page 1 / 1



Budget Allocation Module

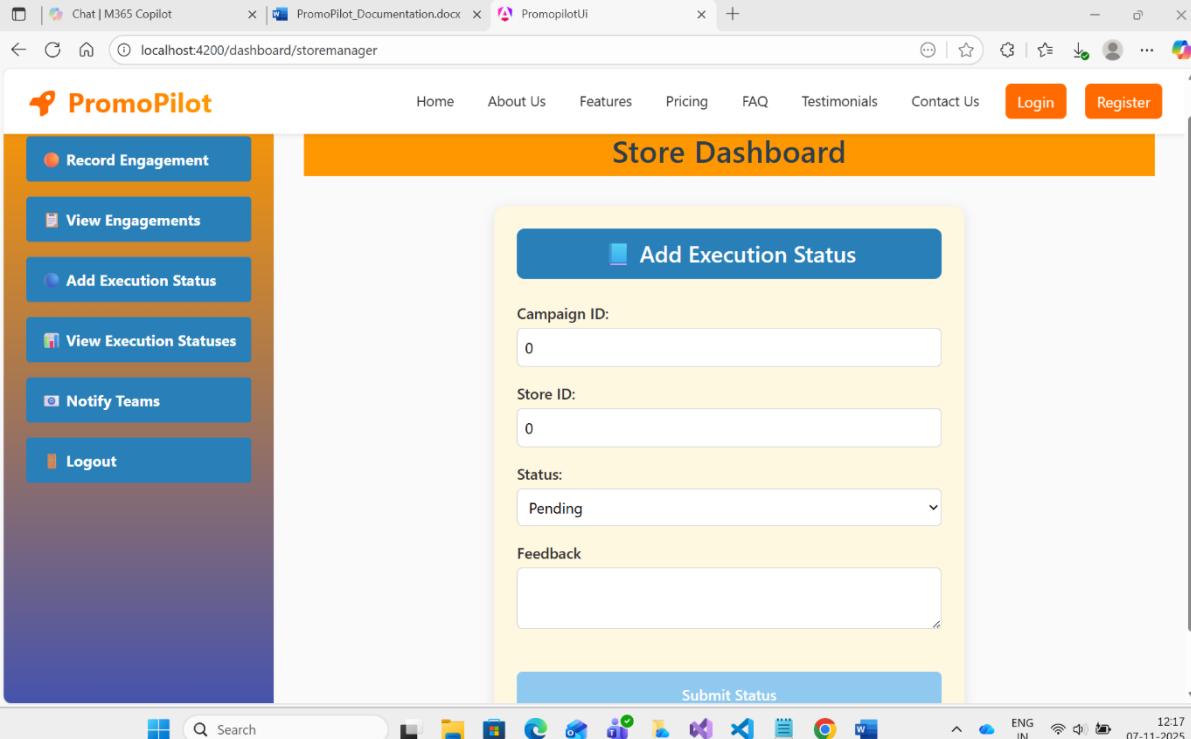
The screenshot shows a web browser window for the 'PromoPilot' application. The title bar reads 'PromopilotUi' and 'localhost:4200/dashboard/finance'. The header includes links for Home, About Us, Features, Pricing, FAQ, Testimonials, Contact Us, Login, and Register. On the left, a sidebar menu has items: 'Allocate Budget' (selected), 'View Budgets', 'Notify Teams', and 'Logout'. The main content area is titled 'Finance Dashboard' and contains a form titled 'Allocate Budget' with fields for Campaign ID, Store ID, Allocated Amount, and Spent Amount, each with an example value ('e.g. 7', 'e.g. 102', 'e.g. 10000', 'e.g. 7000'). A large orange button at the bottom right of the form says 'Allocate Budget'. The taskbar at the bottom shows various application icons.



Engagement Tracking Module

The screenshot shows a web browser window for the 'PromoPilot' application. The title bar reads 'PromopilotUi' and 'localhost:4200/dashboard/storemanager'. The header includes links for Home, About Us, Features (selected), Pricing, FAQ, Testimonials, Contact Us, Login, and Register. On the left, a sidebar menu has items: 'Record Engagement' (selected), 'View Engagements', 'Add Execution Status', 'View Execution Statuses', 'Notify Teams', and 'Logout'. The main content area is titled 'Store Dashboard' and contains a form titled 'Record Engagement' with fields for Campaign ID, Customer ID, Redemption Count, and Purchase Value (₹), each with an example value ('0', '0', '0', '0'). A large orange button at the bottom right of the form says 'Record Engagement'. The taskbar at the bottom shows various application icons.

Execution Status Module

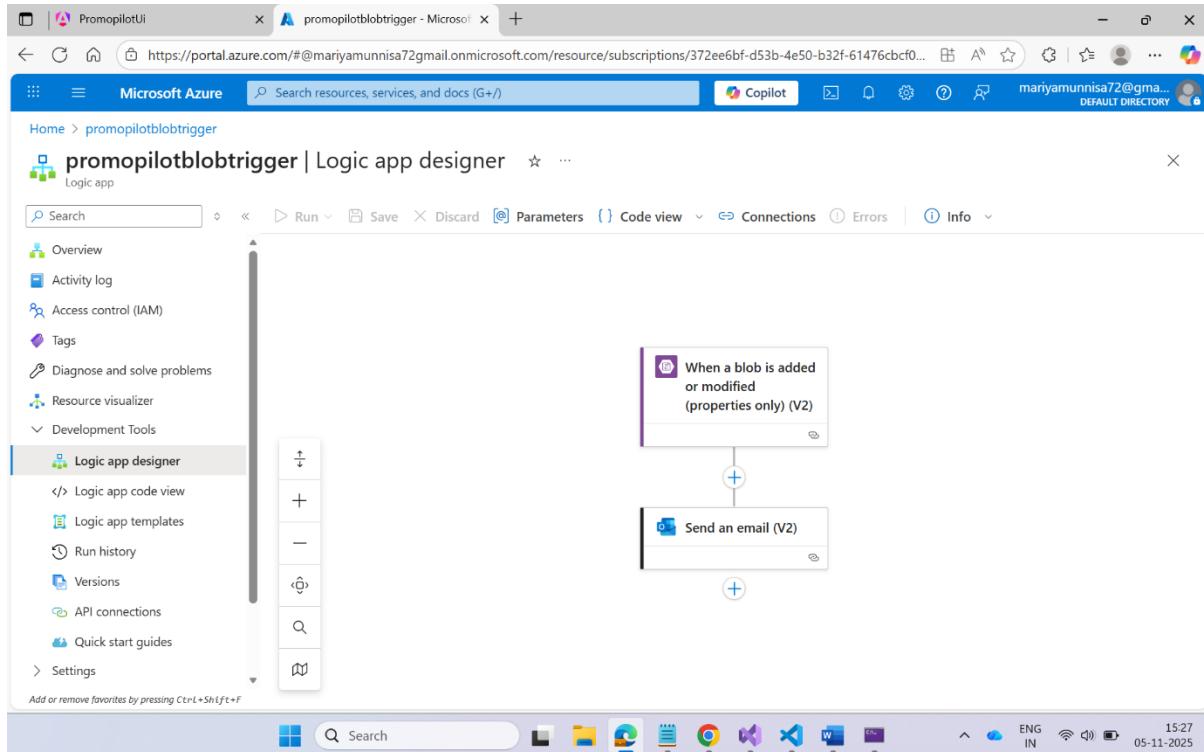


The screenshot shows a web-based application interface for managing store engagement status. On the left, there's a sidebar with various buttons: 'Record Engagement', 'View Engagements', 'Add Execution Status' (which is currently selected and highlighted in blue), 'View Execution Statuses', 'Notify Teams', and 'Logout'. The main content area is titled 'Store Dashboard' and features a form titled 'Add Execution Status'. This form includes fields for 'Campaign ID' (containing '0'), 'Store ID' (containing '0'), 'Status' (a dropdown menu set to 'Pending'), and a 'Feedback' text area. At the bottom of the form is a 'Submit Status' button. The browser's address bar shows 'localhost:4200/dashboard/storemanager'. The taskbar at the bottom of the screen displays several pinned icons, including Microsoft Office applications like Word, Excel, and PowerPoint, as well as other common tools.

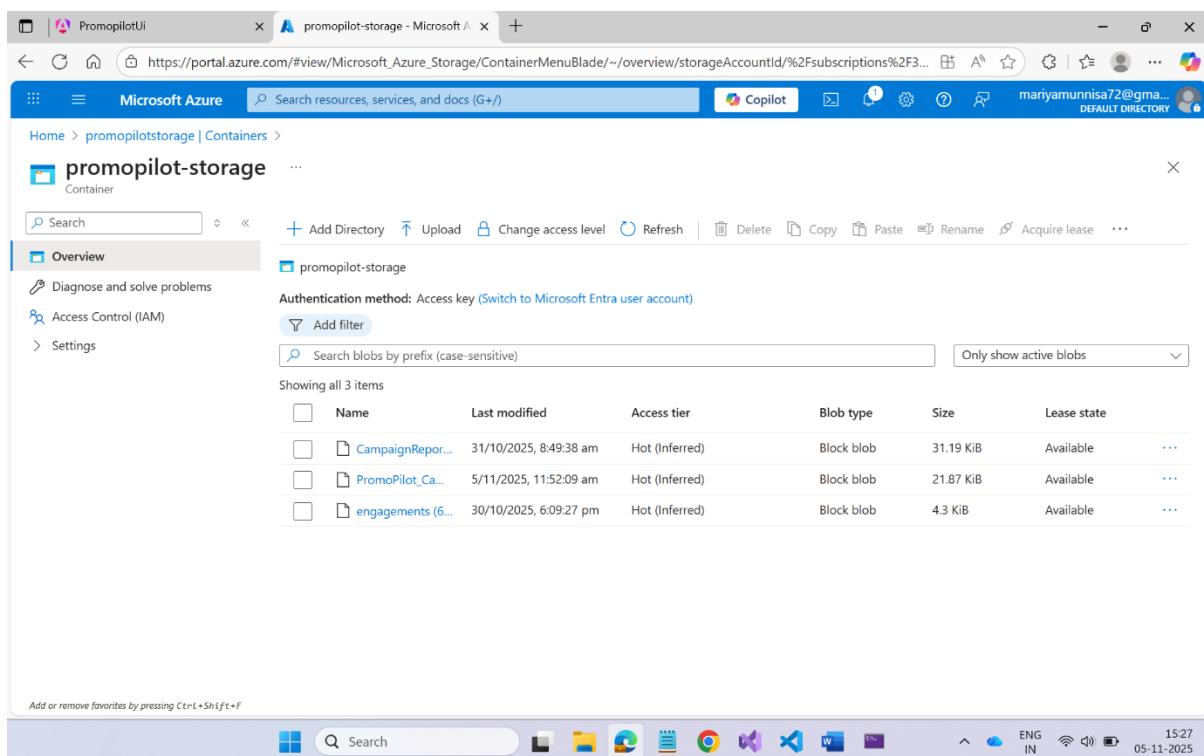
4. Azure Services Used

| Service | Purpose |
|-------------------------------------|---|
| Blob Storage | Stores exported reports (CSV, Excel, PDF) |
| ACS (Communication Services) | Sends welcome emails |
| Logic Apps | Sends email when blob is added |

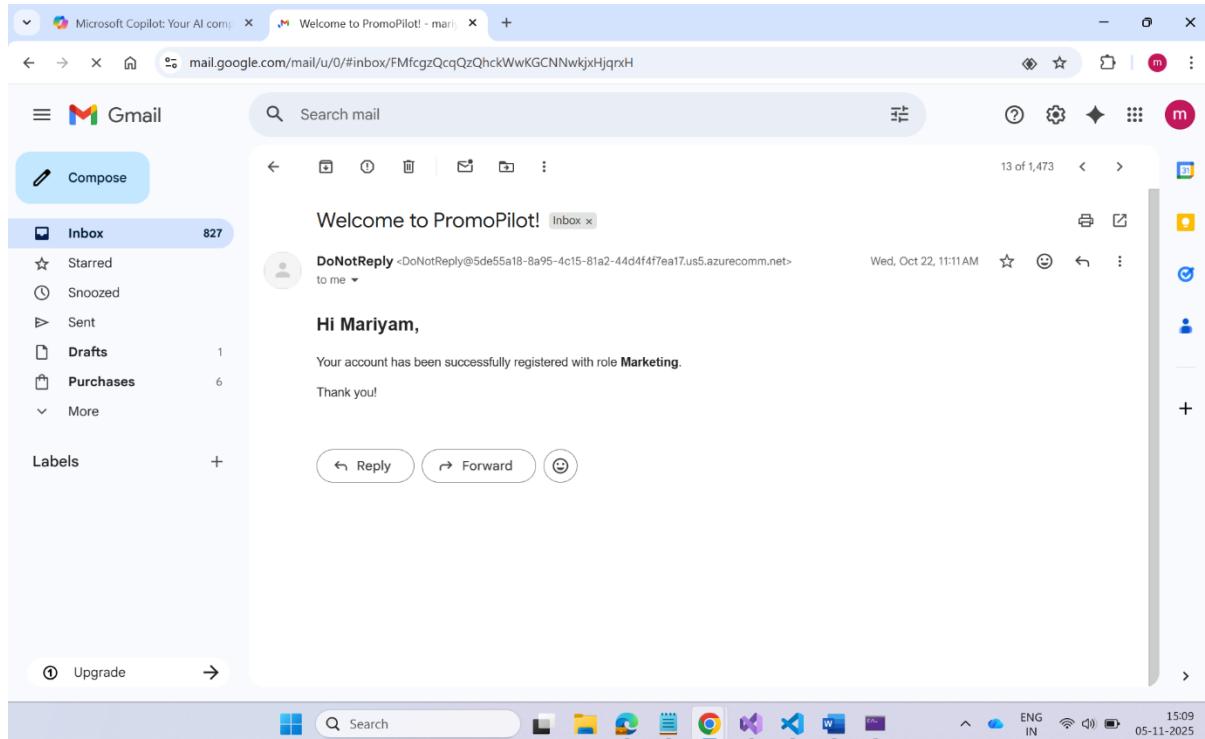
Azure Logic App Designer



Azure Blob Container View



ACS Welcome Email



5. Frontend Code Highlights

login.component.ts

```
login() {  
  
    this.authService.login(this.loginForm.value).subscribe({  
  
        next: (res) => {  
  
            this.router.navigate([`/dashboard/${res.role}`]);  
  
        },  
  
        error: (err) => {  
  
            this.errorMessage = 'Invalid credentials';  
  
        }  
  
    });  
}
```

Purpose: Role-based routing after login, with error feedback.

register.component.ts

```
register() {  
  
  const payload = { ...this.form.value, role: this.selectedRole };  
  
  this.authService.register(payload).subscribe({  
  
    next: () => this.router.navigate(['/login']),  
  
    error: () => this.errorMessage = 'Registration failed'  
  
  });  
  
}
```

Purpose: Role selection and ACS email trigger integration.

home.component.ts

```
home() {  
  
  loadDashboard() {  
  
    this.dashboardService.getSummary().subscribe({  
  
      next: (data) => this.dashboardData = data,  
  
      error: () => this.errorMessage = 'Failed to load dashboard'  
  
    });  
  
  }  
  
}
```

Purpose: User landing page with welcome message, quick navigation (trial, features), and campaign performance overview.

marketing-dashboard.component.ts

```
exportReport(type: 'pdf') {  
  
  this.reportService.export(type).subscribe(blob => {  
  
    this.blobService.upload(blob, `${type}-report`);  
  
  });  
  
}
```

```
}
```

Purpose: Export logic and blob upload for report sharing.

finance-dashboard.component.ts

```
exportBudget(type: 'csv' | 'excel') {  
  this.budgetService.export(type).subscribe(blob => {  
    this.blobService.upload(blob, `budget-${type}-report`);  
  });  
}
```

Purpose: Triggers export of budget data and uploads the file to Azure Blob Storage.

StoreManager-dashboard.component.ts

```
exportEngagement(type: 'csv' | 'excel') {  
  this.engagementService.export(type).subscribe(blob => {  
    this.blobService.upload(blob, `engagement-${type}-report`);  
  });  
}
```

Purpose: Triggers export of engagement data and uploads the file to Azure Blob Storage.

6. Backend Code Highlights

AuthController.cs

```
[HttpPost("login")]  
public async Task<IActionResult> Login(LoginDto dto) {  
  var result = await _authService.LoginAsync(dto);  
  return result.IsSuccess ? Ok(result.Data) : Unauthorized(result.Error);  
}
```

Purpose: Secure login with JWT and role-based response.

AuthService.cs

```
public async Task<AuthResult> LoginAsync(LoginDto dto) {  
    // Validate user, generate token, return role  
}  
}
```

Purpose: Token generation and role extraction.

CampaignController.cs

```
[HttpPost("create")]  
public async Task<IActionResult> Create(CampaignDto dto) {  
    await _campaignService.CreateAsync(dto);  
    return Ok();  
}
```

Purpose: Campaign creation endpoint.

CampaignService.cs

```
public async Task<CampaignDto> AddAsync(CampaignDto dto){  
    // Validate and persist new campaign details, return created campaign DTO  
}
```

Purpose: Campaign creation and persistence.

BudgetController.cs

```
[HttpPost("allocate")]  
public async Task<IActionResult> Allocate(BudgetDto dto) {  
    await _budgetService.AllocateAsync(dto);  
    return Ok();  
}
```

Purpose: Budget allocation endpoint.

BudgetService.cs

```
public async Task AllocateBudgetAsync(Budget budget)
```

Purpose: Budget allocation and tracking.

EngagementController.cs

```
[HttpPost("Track")]
public async Task<IActionResult> Track(EngagementDto dto) {
    await _engagementService.TrackAsync(dto);
    return Ok();
}
```

Purpose: Engagement tracking endpoint.

EngagementService.cs

```
public async Task AddAsync(Engagement engagement)
// Record customer engagement metrics, store interaction details, redemption count
}
```

Purpose: Engagement Tracking and redemption count tracking.

ExecutionStatusController.cs

```
[HttpPost(" Add ")]
public async Task<IActionResult> Add(ExecutionStatusDto dto) {
    await _executionStatusService.AddAsync(dto);
    return Ok();
}
```

Purpose: Execution Status Tracking and Feedback endpoint.

ExecutionStatusService.cs

```
public async Task AddAsync(ExecutionStatus status){  
    // Add execution status updates, track workflow progress  
}
```

Purpose: Execution Status monitoring and Feedback management.

BlobController.cs

```
[HttpPost("upload")]  
  
public async Task<IActionResult> Upload(IFormFile file) {  
  
    var result = await _blobService.UploadAsync(file);  
  
    return Ok(result);  
}
```

Purpose: Upload exported reports to Azure Blob Storage.

BlobStorageService.cs

```
public async Task<string> UploadAsync(IFormFile file) {  
  
    var container = _blobClient.GetContainerReference("reports");  
  
    var blob = container.GetBlockBlobReference(file.FileName);  
  
    await blob.UploadFromStreamAsync(file.OpenReadStream());  
  
    return blob.Uri.ToString();  
}
```

Purpose: Uploads exported reports to Azure Blob Storage and returns the blob URL.