

# AWS Amplify Backend Implementation Plan for ABU Alumni Platform

## 1. Project Setup & Authentication

### Initial Setup

```
bash
```

```
# Initialize Amplify in your existing frontend project
```

```
amplify init
```

```
# Add authentication
```

```
amplify add auth
```

```
# Configure auth with the settings we defined earlier
```

```
? Do you want to use the default authentication and security configuration? Manual configur
```

```
? Select the authentication/authorization services that you want to use: User Sign-Up, Sign
```

```
? Provide a friendly name for your resource that will be used to label this category in the
```

```
? How do you want users to be able to sign in? Email, Username
```

```
? Do you want to configure advanced settings? Yes
```

```
...
```

```
# Configure user groups for Alumni, Admins, etc.
```



### Auth Functions

Create Lambda triggers for:

- Pre-signup validation
- Post-confirmation (to create initial user profile)
- Custom message handling for branded emails

## 2. GraphQL API Setup

```
bash
```

```
# Add API
```

```
amplify add api
```

```
# Choose GraphQL, with Amazon Cognito user pool for authorization type
```

```
? Please select from one of the below mentioned services: GraphQL
```

```
? Provide API name: abuAlumniAPI
```

```
? Choose the default authorization type for the API: Amazon Cognito User Pool
```

```
? Do you want to configure advanced settings for the GraphQL API: Yes
```

```
? Configure additional auth types? Yes
```

```
? Choose the additional authorization types you want to configure for the API: IAM, API key
```



Import the GraphQL schema we defined earlier.

### 3. Storage Configuration

```
bash
```

```
# Add storage for user files (profile pictures, event images, etc.)
```

```
amplify add storage
```

```
# Configure S3 with appropriate access patterns
```

```
? Please select from one of the below mentioned services: Content
```

```
? Please provide a friendly name for your resource that will be used to label this category
```

```
? Please provide bucket name: abu-alumni-storage
```

```
? Who should have access: Auth users only
```

```
? What kind of access do you want for Authenticated users? create/update, read, delete
```



### 4. Functions Implementation

Create Lambda functions for:

#### Alumni Search Function

```
bash
```

```
amplify add function
```

```
? Provide a friendly name for your resource: searchAlumniFunction
```

```
? Provide the AWS Lambda function name: searchAlumniFunction
```

```
? Choose the runtime that you want to use: NodeJS
```

```
? Choose the function template that you want to use: Hello World
```

Implementation will include:

- Elasticsearch integration for advanced alumni search
- Filtering by graduation year, department, location, etc.

## Dashboard Statistics Function

```
bash
```

```
amplify add function
```

```
? Provide a friendly name for your resource: getDashboardStatsFunction
```

```
? Provide the AWS Lambda function name: getDashboardStatsFunction
```

```
? Choose the runtime that you want to use: NodeJS
```

```
? Choose the function template that you want to use: Hello World
```

Implementation will aggregate data across:

- User counts and activity
- Donation metrics
- Event participation
- Group engagement

## Payment Processing Functions

```
bash
```

```
amplify add function
```

```
? Provide a friendly name for your resource: getPaymentIntentFunction
```

```
? Provide the AWS Lambda function name: getPaymentIntentFunction
```

```
? Choose the runtime that you want to use: NodeJS
```

```
? Choose the function template that you want to use: Hello World
```

Implementation will:

- Integrate with PayStack, Stripe, or PayPal
- Handle payment intents securely
- Process international payments
- Create donation records

## Notification Function

```
bash
```

```
amplify add function
```

```
? Provide a friendly name for your resource: sendBulkNotificationsFunction
? Provide the AWS Lambda function name: sendBulkNotificationsFunction
? Choose the runtime that you want to use: NodeJS
? Choose the function template that you want to use: Hello World
```

Implementation will handle:

- Push notifications via Amazon SNS
- Email notifications via Amazon SES
- In-app notifications saved to database

## 5. Hosting Configuration

```
bash
```

```
# Add hosting for your web frontend
amplify add hosting
```

```
# Configure CloudFront distribution for global access
? Select the plugin module to execute: Hosting with Amplify Console
? Choose a type: Manual deployment
```

## 6. CI/CD Setup

Configure Amplify Console CI/CD pipeline:

- Connect to your GitHub repository
- Set up environment variables
- Configure build settings for web and backend
- Set up branch-based deployments (dev, staging, production)

## 7. Security & Compliance Implementation

### Data Protection

- Implement field-level encryption for sensitive data

- Set up AWS WAF for API protection
- Configure VPC for database isolation

## **Compliance**

- Set up CloudWatch Logs for audit trails
- Implement PCI-DSS compliant patterns for payment handling
- Configure GDPR-compliant data retention policies

## **8. Testing Plan**

### **API Testing**

- Create collections in Postman for all API endpoints
- Write automated tests using Jest for Lambda functions
- Set up integration tests for complete user flows

### **Security Testing**

- Perform penetration testing on APIs
- Conduct IAM policy reviews
- Test authentication flows for vulnerabilities

## **9. Deployment Strategy**

### **Phased Rollout**

1. Deploy Authentication & Core User Management
2. Deploy Profile & Networking Features
3. Deploy Events & Content Management
4. Deploy Donation System
5. Deploy Community Features (Groups & Projects)

### **For Each Phase:**

- Create feature branch
- Implement and test locally
- Deploy to dev environment
- Conduct QA testing
- Deploy to staging for stakeholder review

- Deploy to production

## **10. Post-Launch Monitoring**

### **Observability Setup**

- Configure CloudWatch dashboards for key metrics
- Set up alarms for performance and error thresholds
- Implement X-Ray for tracing API calls and Lambda executions

### **Analytics Integration**

- Set up Amplify Analytics category
- Implement event tracking for key user actions
- Create dashboards for user engagement metrics

## **11. Backup & Disaster Recovery**

### **Data Protection**

- Configure automated backups for DynamoDB tables
- Set up cross-region replication for S3 buckets
- Create restoration procedures documentation

### **Failover Strategy**

- Design multi-region deployment for high availability
- Document disaster recovery procedures
- Schedule regular restoration drills