

AlumNet API Documentation

Base URL

```
http://localhost:4000/api
```

Authentication

Most endpoints require authentication using JWT Bearer token in the Authorization header:

```
Authorization: Bearer <your-jwt-token>
```

Authentication Endpoints

1. Register User

POST /auth/register

Request Body:

```
{
  "name": "John Doe",
  "email": "john@college.edu",
  "password": "password123",
  "role": "student", // optional: "student" | "alumni"
  "graduationYear": 2023, // optional
  "department": "Computer Science", // optional
  "batch": 2019, // required for alumni
  "course": "B.Tech", // required for alumni
  "currentJob": "Software Engineer" // optional for alumni
}
```

Response (201 Created):

```
{
  "user": {
    "id": "64f1a2b3c4d5e6f7g8h9i0j1",
    "name": "John Doe",
    "email": "john@college.edu",
    "role": "student"
  },
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

2. Login User

POST /auth/login

Request Body:

```
{
  "email": "john@college.edu",
  "password": "password123"
}
```

Response (200 OK):

```
{
  "user": {
    "id": "64f1a2b3c4d5e6f7g8h9i0j1",
    "name": "John Doe",
    "email": "john@college.edu",
    "role": "student",
    "avatarUrl": "https://cdn-icons-png.flaticon.com/128/3177/3177440.png"
  },
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

3. Get Current User

GET /auth/me

Response (200 OK):

```
{
  "user": {
    "id": "64f1a2b3c4d5e6f7g8h9i0j1",
    "name": "John Doe",
    "email": "john@college.edu",
    "role": "student",
    "graduationYear": 2023,
    "department": "Computer Science",
    "avatarUrl": "https://cdn-icons-png.flaticon.com/128/3177/3177440.png",
    "verified": true,
    "isActive": true,
    "createdAt": "2023-09-01T10:00:00.000Z",
    "updatedAt": "2023-09-01T10:00:00.000Z"
  }
}
```

4. Update Current User

PUT /auth/me

Request Body:

```
{
  "name": "John Smith",
  "graduationYear": 2024,
  "department": "Information Technology",
  "batch": 2020, // only for alumni
  "course": "M.Tech", // only for alumni
  "currentJob": "Senior Developer" // only for alumni
}
```

Response (200 OK):

```
{
  "user": {
    "id": "64f1a2b3c4d5e6f7g8h9i0j1",
    "name": "John Smith",
    "email": "john@college.edu",
    "role": "student"
  }
}
```

5. Forgot Password

POST /auth/forgot-password

Request Body:

```
{
  "email": "john@college.edu"
}
```

Response (200 OK):

```
{
  "message": "Password reset token generated",
  "resetToken": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0u1v2w3x4y5z6"
}
```

6. Reset Password

POST /auth/reset-password

Request Body:

```
{
  "token": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0u1v2w3x4y5z6",
  "password": "newpassword123"
}
```

Response (200 OK):

```
{
  "message": "Password has been reset successfully"
}
```

Posts Endpoints

1. Get All Posts

GET /posts

Query Parameters:

- page (optional): Page number (default: 1)
- limit (optional): Posts per page (default: 10)
- search (optional): Search in title and content

Example: /posts?page=1&limit=5&search=javascript

Response (200 OK):

```
{
  "posts": [
    {
      "_id": "64f1a2b3c4d5e6f7g8h9i0j2",
      "title": "JavaScript Best Practices",
      "content": "Here are some JavaScript best practices...",
      "imageUrl": "/uploads/javascript-guide-1693567890123.jpg",
      "author": {
        "_id": "64f1a2b3c4d5e6f7g8h9i0j1",
        "name": "John Doe",
        "role": "alumni"
      },
      "likes": ["64f1a2b3c4d5e6f7g8h9i0j1", "64f1a2b3c4d5e6f7g8h9i0j3"],
      "comments": [
        {
          "_id": "64f1a2b3c4d5e6f7g8h9i0j4",
          "user": "64f1a2b3c4d5e6f7g8h9i0j1",
          "text": "Great post!",
          "createdAt": "2023-09-01T11:00:00.000Z"
        }
      ],
      "likesCount": 2,
      "commentsCount": 1,
      "createdAt": "2023-09-01T10:30:00.000Z",
      "updatedAt": "2023-09-01T10:30:00.000Z"
    }
  ],
  "pagination": {
    "currentPage": 1,
    "totalPages": 3,
    "totalPosts": 25,
    "hasNext": true,
    "hasPrev": false
  },
  "search": "javascript"
}
```

2. Get Single Post

GET /posts/:id

Response (200 OK):

```
{
  "_id": "64f1a2b3c4d5e6f7g8h9i0j2",
  "title": "JavaScript Best Practices",
  "content": "Here are some JavaScript best practices...",
  "imageUrl": "/uploads/javascript-guide-1693567890123.jpg",
  "author": {
    "_id": "64f1a2b3c4d5e6f7g8h9i0j1",
    "name": "John Doe",
    "role": "alumni"
  },
  "likes": ["64f1a2b3c4d5e6f7g8h9i0j1", "64f1a2b3c4d5e6f7g8h9i0j3"],
  "comments": [
    {
      "_id": "64f1a2b3c4d5e6f7g8h9i0j4",
      "user": "64f1a2b3c4d5e6f7g8h9i0j1",
      "text": "Great post!",
      "createdAt": "2023-09-01T11:00:00.000Z"
    }
  ],
  "likesCount": 2,
  "commentsCount": 1,
  "createdAt": "2023-09-01T10:30:00.000Z",
  "updatedAt": "2023-09-01T10:30:00.000Z"
}
```

3. Create Post

POST /posts (Alumni/Admin only)

Request Body (multipart/form-data):

```
{
  "title": "My New Post",
  "content": "This is the content of my post",
  "image": "file" // optional image file
}
```

Response (201 Created):

```
{
  "_id": "64f1a2b3c4d5e6f7g8h9i0j2",
  "title": "My New Post",
  "content": "This is the content of my post",
  "imageUrl": "/uploads/my-post-1693567890123.jpg",
  "author": "64f1a2b3c4d5e6f7g8h9i0j1",
  "likes": [],
  "comments": [],
  "createdAt": "2023-09-01T10:30:00.000Z",
  "updatedAt": "2023-09-01T10:30:00.000Z"
}
```

4. Update Post

PATCH /posts/:id (Author/Admin only)

Request Body (multipart/form-data):

```
{
  "title": "Updated Post Title",
  "content": "Updated content",
  "image": "file" // optional new image file
}
```

Response (200 OK):

```
{
  "_id": "64f1a2b3c4d5e6f7g8h9i0j2",
  "title": "Updated Post Title",
  "content": "Updated content",
  "imageUrl": "/uploads/updated-post-1693567890123.jpg",
  "author": "64f1a2b3c4d5e6f7g8h9i0j1",
  "likes": ["64f1a2b3c4d5e6f7g8h9i0j1"],
  "comments": [],
  "createdAt": "2023-09-01T10:30:00.000Z",
  "updatedAt": "2023-09-01T11:00:00.000Z"
}
```

5. Delete Post

DELETE /posts/:id (Author/Admin only)

Response (204 No Content):

(Empty response body)

6. Like/Unlike Post

POST /posts/:id/like

Response (200 OK):

```
{
  "liked": true,
  "likes": 5
}
```

7. Comment on Post

POST /posts/:id/comment

Request Body:

```
{
  "text": "This is a great post!"
}
```

Response (201 Created):

```
{
  "_id": "64f1a2b3c4d5e6f7g8h9i0j4",
  "user": "64f1a2b3c4d5e6f7g8h9i0j1",
  "text": "This is a great post!",
  "createdAt": "2023-09-01T11:00:00.000Z"
}
```

Events Endpoints

1. Get All Events

GET /events

Query Parameters:

- page (optional): Page number (default: 1)
- limit (optional): Events per page (default: 10)

Response (200 OK):

```
{
  "events": [
    {
      "_id": "64f1a2b3c4d5e6f7g8h9i0j5",
      "title": "Alumni Meet 2023",
      "description": "Annual alumni gathering",
      "date": "2023-12-15T18:00:00.000Z",
      "location": "College Auditorium",
      "organizer": {
        "_id": "64f1a2b3c4d5e6f7g8h9i0j1",
        "name": "John Doe"
      },
      "attendees": ["64f1a2b3c4d5e6f7g8h9i0j1", "64f1a2b3c4d5e6f7g8h9i0j3"],
      "createdAt": "2023-09-01T10:00:00.000Z",
      "updatedAt": "2023-09-01T10:00:00.000Z"
    }
  ],
  "pagination": {
    "currentPage": 1,
    "totalPages": 2,
    "totalEvents": 15,
    "hasNext": true,
    "hasPrev": false
  }
}
```

2. Get Single Event

GET /events/:id

Response (200 OK):

```
{
  "_id": "64f1a2b3c4d5e6f7g8h9i0j5",
  "title": "Alumni Meet 2023",
  "description": "Annual alumni gathering",
  "date": "2023-12-15T18:00:00.000Z",
  "location": "College Auditorium",
  "organizer": {
    "_id": "64f1a2b3c4d5e6f7g8h9i0j1",
    "name": "John Doe"
  },
  "attendees": ["64f1a2b3c4d5e6f7g8h9i0j1", "64f1a2b3c4d5e6f7g8h9i0j3"],
  "createdAt": "2023-09-01T10:00:00.000Z",
  "updatedAt": "2023-09-01T10:00:00.000Z"
}
```

3. Create Event

POST /events

Request Body:


```
{
  "title": "Tech Conference 2023",
  "description": "Annual technology conference",
  "date": "2023-11-20T09:00:00.000Z",
  "location": "Convention Center"
}
```

Response (201 Created):

```
{
  "_id": "64f1a2b3c4d5e6f7g8h9i0j6",
  "title": "Tech Conference 2023",
  "description": "Annual technology conference",
  "date": "2023-11-20T09:00:00.000Z",
  "location": "Convention Center",
  "organizer": "64f1a2b3c4d5e6f7g8h9i0j1",
  "attendees": [],
  "createdAt": "2023-09-01T10:00:00.000Z",
  "updatedAt": "2023-09-01T10:00:00.000Z"
}
```

4. Update Event

PATCH /events/:id (Organizer/Admin only)

Request Body:

```
{
  "title": "Updated Tech Conference 2023",
  "description": "Updated description",
  "date": "2023-11-25T09:00:00.000Z",
  "location": "New Convention Center"
}
```

Response (200 OK):

```
{
  "_id": "64f1a2b3c4d5e6f7g8h9i0j6",
  "title": "Updated Tech Conference 2023",
  "description": "Updated description",
  "date": "2023-11-25T09:00:00.000Z",
  "location": "New Convention Center",
  "organizer": "64f1a2b3c4d5e6f7g8h9i0j1",
  "attendees": ["64f1a2b3c4d5e6f7g8h9i0j1"],
  "createdAt": "2023-09-01T10:00:00.000Z",
  "updatedAt": "2023-09-01T11:00:00.000Z"
}
```

5. Delete Event

DELETE /events/:id (Organizer/Admin only)

Response (204 No Content):

(Empty response body)

6. Attend Event

POST /events/:id/attend

Response (200 OK):

```
{  
  "attendees": 3  
}
```

7. Leave Event

POST /events/:id/leave

Response (200 OK):

```
{  
  "attendees": 2  
}
```

User Endpoints

1. Get Public Alumni

GET /users/alumni

Query Parameters:

- q (optional): Search query
- department (optional): Filter by department
- course (optional): Filter by course
- job (optional): Filter by current job
- batch (optional): Filter by batch year
- graduationYear (optional): Filter by graduation year

Example: /users/alumni?q=software&department=Computer Science&batch=2019

Response (200 OK):

```
{
  "alumni": [
    {
      "_id": "64f1a2b3c4d5e6f7g8h9i0j1",
      "name": "John Doe",
      "email": "john@college.edu",
      "role": "alumni",
      "graduationYear": 2019,
      "department": "Computer Science",
      "avatarUrl": "https://cdn-icons-png.flaticon.com/128/3177/3177440.png",
      "batch": 2019,
      "course": "B.Tech",
      "currentJob": "Software Engineer",
      "verified": true
    }
  ]
}
```

Profile Endpoints

1. Get My Profile

GET /profile/me

Response (200 OK):

```
{
  "user": {
    "_id": "64f1a2b3c4d5e6f7g8h9i0j1",
    "name": "John Doe",
    "email": "john@college.edu",
    "role": "student",
    "graduationYear": 2023,
    "department": "Computer Science",
    "avatarUrl": "https://cdn-icons-png.flaticon.com/128/3177/3177440.png",
    "batch": null,
    "course": null,
    "currentJob": null,
    "verified": true,
    "isActive": true,
    "createdAt": "2023-09-01T10:00:00.000Z",
    "updatedAt": "2023-09-01T10:00:00.000Z"
  }
}
```

2. Update My Profile

PUT /profile/me

Request Body:

```
{
  "name": "John Smith",
  "graduationYear": 2024,
  "department": "Information Technology"
}
```

Response (200 OK):

```
{
  "user": {
    "id": "64f1a2b3c4d5e6f7g8h9i0j1",
    "name": "John Smith",
    "email": "john@college.edu",
    "role": "student"
  }
}
```

3. Upload Avatar

POST /profile/me/avatar

Request Body (multipart/form-data):

```
avatar: <image file>
```

Response (200 OK):

```
{
  "avatarUrl": "/uploads/avatar-1693567890123.jpg"
}
```

4. Get Public Profile by ID

GET /profile/:id

Response (200 OK):

```
{
  "user": {
    "id": "64f1a2b3c4d5e6f7g8h9i0j1",
    "name": "John Doe",
    "email": "john@college.edu",
    "role": "alumni",
    "graduationYear": 2019,
    "department": "Computer Science",
    "avatarUrl": "https://cdn-icons-png.flaticon.com/128/3177/3177440.png",
    "batch": 2019,
    "course": "B.Tech",
    "currentJob": "Software Engineer",
    "verified": true
  }
}
```

Admin Endpoints

1. List Users

GET /admin/users (Admin only)

Query Parameters:

- role (optional): Filter by role
- verified (optional): Filter by verification status
- q (optional): Search query

Response (200 OK):

```
{
  "users": [
    {
      "_id": "64f1a2b3c4d5e6f7g8h9i0j1",
      "name": "John Doe",
      "email": "john@college.edu",
      "role": "alumni",
      "graduationYear": 2019,
      "department": "Computer Science",
      "avatarUrl": "https://cdn-icons-png.flaticon.com/128/3177/3177440.png",
      "batch": 2019,
      "course": "B.Tech",
      "currentJob": "Software Engineer",
      "verified": true,
      "isActive": true,
      "createdAt": "2023-09-01T10:00:00.000Z",
      "updatedAt": "2023-09-01T10:00:00.000Z"
    }
  ]
}
```

2. Verify Alumni

POST /admin/users/:id/verify (Admin only)

Response (200 OK):

```
{
  "message": "Alumni verified",
  "user": {
    "id": "64f1a2b3c4d5e6f7g8h9i0j1",
    "name": "John Doe",
    "email": "john@college.edu",
    "role": "alumni",
    "verified": true
  }
}
```

3. Set User Active Status

PATCH /admin/users/:id/active (Admin only)

Request Body:

```
{  
  "isActive": false  
}
```

Response (200 OK):

```
{  
  "message": "User state updated",  
  "user": {  
    "id": "64f1a2b3c4d5e6f7g8h9i0j1",  
    "isActive": false  
  }  
}
```

4. Delete User

DELETE /admin/users/:id (Admin only)

Response (200 OK):

```
{  
  "message": "User deleted"  
}
```

Health Check

Health Check

GET /health

Response (200 OK):

```
{  
  "ok": true,  
  "service": "AlumniConnect API"  
}
```

Error Responses

400 Bad Request

```
{
  "message": "Validation failed",
  "errors": [
    {
      "msg": "Name is required",
      "param": "name",
      "location": "body"
    }
  ]
}
```

401 Unauthorized

```
{
  "message": "Unauthorized"
}
```

403 Forbidden

```
{
  "message": "Forbidden"
}
```

404 Not Found

```
{
  "message": "Post not found"
}
```

409 Conflict

```
{
  "message": "Email already in use"
}
```

429 Too Many Requests

```
{
  "message": "Too many requests from this IP, please try again later."
}
```

500 Internal Server Error

```
{
  "message": "Internal Server Error"
}
```

Status Codes

Code	Description
------	-------------

200 OK - Request successful
201 Created - Resource created successfully
204 No Content - Request successful, no content returned
400 Bad Request - Invalid request data
401 Unauthorized - Authentication required
403 Forbidden - Insufficient permissions
404 Not Found - Resource not found
409 Conflict - Resource already exists
429 Too Many Requests - Rate limit exceeded
500 Internal Server Error - Server error

Authentication & Authorization

User Roles

- **student:** Can view content, attend events, comment on posts
- **alumni:** Can create posts, organize events, full access to alumni features
- **admin:** Full system access, user management, content moderation

Rate Limiting

- **General endpoints:** 100 requests per 15 minutes
- **Auth endpoints:** 5 requests per 15 minutes

File Upload

- **Supported formats:** PNG, JPEG, GIF, WebP
 - **Maximum size:** 2MB
 - **Validation:** Both MIME type and file extension checking
-

Notes

- All timestamps are in ISO 8601 format (UTC)
- Object IDs are MongoDB ObjectIds (24 character hex strings)
- Password reset tokens expire after 10 minutes
- Alumni accounts require admin verification to create posts
- File uploads are stored in the /uploads directory
- CORS is enabled for cross-origin requests
- Input sanitization prevents NoSQL injection attacks