Category of service  
1.Health care ->disease->eye test,choleara test….  
2.scheam->gram yogona, krisak , Balika siksha yogona  
3.Education->vocational , welding,tailoring  
4.  
Certainly! Let’s break down your **Digital E Gram Panchayat** project and explain **how to implement it using Django (backend) and React (frontend)** in a clear, modular way.

**🧩 PROBLEM OVERVIEW (Elaborated)**

The **Digital E Gram Panchayat System** is a web application designed to:

* **Digitalize** and **simplify** the delivery of government services at the village level.
* Allow **citizens** (Users) to **register, search, and apply** for services/schemes.
* Let **Officers/Admins** manage services and update the status of applications.
* Allow **Staff** to help process and update application statuses.

**👥 ROLES & RESPONSIBILITIES**

| **Role** | **Responsibilities** |
| --- | --- |
| **User** | Register, login, search services, apply, view status/profile |
| **Staff** | Login, view service applications, update status |
| **Officer/Admin** | Login, create/edit/delete services, manage applications and users |

**🏗️ MODULES AND FUNCTIONALITIES**

**1. User**

* Register / Login
* View/Search services
* Apply for services
* View application status
* View/edit profile

**2. Staff**

* Login
* View service applications
* Update application statuses (e.g., “Approved”, “In Progress”, “Rejected”)

**3. Admin/Officer**

* Login
* Create/Update/Delete services
* View and update user applications
* Manage staff

**🛠 TECHNOLOGY STACK**

**Backend (Django)**

* Django + Django REST Framework for APIs
* PostgreSQL or SQLite (development) as database
* Token-based authentication (Simple JWT)
* Django Admin for managing data (optional)

**Frontend (React)**

* React + Axios for API calls
* React Router for routing
* Tailwind CSS or Bootstrap for UI
* Role-based access control in the frontend

**🔁 DATA FLOW**

1. **User registers/logs in** → Auth token issued from Django.
2. **User applies for a service** → API call from React to Django to submit form.
3. **Staff/Admin views pending applications** → Updates application status.
4. **User checks application status** on their dashboard.

**🔧 DATABASE MODELS (Django ORM)**

python

CopyEdit

# models.py

class User(AbstractUser):

ROLE\_CHOICES = (

('user', 'User'),

('staff', 'Staff'),

('officer', 'Officer'),

)

role = models.CharField(max\_length=10, choices=ROLE\_CHOICES, default='user')

class Service(models.Model):

name = models.CharField(max\_length=100)

description = models.TextField()

created\_by = models.ForeignKey(User, on\_delete=models.CASCADE)

class Application(models.Model):

user = models.ForeignKey(User, on\_delete=models.CASCADE)

service = models.ForeignKey(Service, on\_delete=models.CASCADE)

status = models.CharField(max\_length=50, default='Pending')

submitted\_at = models.DateTimeField(auto\_now\_add=True)

**📲 API ENDPOINTS (Django REST Framework)**

| **Endpoint** | **Method** | **Role** | **Description** |
| --- | --- | --- | --- |
| /api/register/ | POST | Public | Register user |
| /api/login/ | POST | Public | Login and get JWT |
| /api/services/ | GET | All | List available services |
| /api/services/ | POST | Officer | Create a new service |
| /api/services/:id/ | PUT | Officer | Update a service |
| /api/applications/ | GET | Officer/Staff | View all applications |
| /api/my-applications/ | GET | User | User's application status |
| /api/applications/:id/ | PATCH | Staff/Officer | Update application status |

**🖼️ FRONTEND (React) — MODULE STRUCTURE**

css

CopyEdit

src/

├── components/

│ ├── RegisterForm.js

│ ├── LoginForm.js

│ ├── ServiceList.js

│ ├── ApplyForm.js

│ ├── ApplicationStatus.js

│ └── AdminPanel.js

├── pages/

│ ├── Home.js

│ ├── Dashboard.js

│ ├── Services.js

│ ├── AdminDashboard.js

│ └── StaffDashboard.js

└── App.js

**🛡️ AUTH & ACCESS CONTROL**

* Use JWT tokens stored in localStorage
* Use axios.interceptors to send token in every request
* Protect routes with role-based checks (e.g., only show AdminPanel if role === 'officer')

**🚀 DEPLOYMENT PLAN**

* **Backend (Django)**: Host on **Render**, **Heroku**, or **DigitalOcean**
* **Frontend (React)**: Deploy on **Vercel**, **Netlify**, or **Firebase Hosting**
* Use **PostgreSQL** for production database

**✅ FINAL DELIVERABLES**

1. **GitHub Repo (public)** with:
   * Proper folder structure
   * Modular, clean code
   * Role-based access in frontend
   * API documentation in README.md
2. **README.md** should include:
   * Project overview
   * Setup instructions
   * API endpoints
   * User roles and access flow
   * Sample screenshots or demo link

**🧪 TEST CASE EXAMPLES**

| **Test Case** | **Expected Outcome** |
| --- | --- |
| User registers and logs in | Receives JWT, redirected to dashboard |
| Officer creates a service | Service appears in user-facing service list |
| User applies for a service | Application status shown as "Pending" |
| Staff updates status | User sees updated status in dashboard |

**1. Birth Certificate Application**

* Apply for issuance of a birth certificate.
* Required documents: Hospital report, parent's ID, proof of residence.

**2. Death Certificate Application**

* For reporting and obtaining a certified death certificate.
* Required documents: Hospital/doctor certificate, ID proof of deceased.

**3. Domicile Certificate**

* Certifies a person’s permanent residence in the village.
* Useful for education, job, and government schemes.

**4. Income Certificate**

* Required for scholarships, ration card, and scheme eligibility.
* Requires income proof, Aadhaar, and bank details.

**5. Caste Certificate**

* SC/ST/OBC/General caste certification.
* Documents needed: Parent's caste proof, self ID, and address proof.

**6. Water Connection Request**

* Apply for new household or agricultural water connection.
* Includes site survey and approval from staff.

**7. Sanitation Services**

* Application for waste collection, septic tank cleaning, and drainage services.

**8. Land Ownership or Mutation Request**

* Change ownership due to inheritance or sale.
* Needs legal and land documents.

**9. Electricity Connection**

* Request for new electricity connection in rural areas.
* Approval involves inspection by staff.

**10. Pension Scheme Enrollment**

* For widows, elderly, or differently-abled.
* Needs age proof, medical certificate (if applicable), and income proof.

**11. Village Scheme Registration**

* Enroll in central or state welfare schemes (e.g., Pradhan Mantri Awas Yojana, Ujjwala Yojana).
* Admin updates active schemes and eligibility.

**12. Marriage Registration**

* Apply for a government-recognized marriage certificate.
* Requires witness ID, bride and groom details.

**💼 How Admin Can Manage These Services**

| **Feature** | **Description** |
| --- | --- |
| **Create Service** | Add a new service with title, description, required docs |
| **Edit/Delete Service** | Update or remove outdated services |
| **View Applications** | Check all applications submitted for each service |
| **Approve/Reject** | Change status and add remarks for users to track |
| **Assign to Staff** | Forward requests to relevant staff for action |

**✅ Sample JSON (for API or DB seed)**

json

CopyEdit

{

"title": "Birth Certificate Application",

"description": "Apply for a government-issued birth certificate.",

"required\_documents": [

"Parent ID proof",

"Hospital birth record",

"Proof of residence"

],

"available": true

}

**Categorized Services List**

**1. Health Care**

* **Disease Tests**
  + Eye Test
  + Cholera Test
  + Blood Pressure Check
  + Malaria Screening
* **Preventive Care**
  + Vaccination Drives
  + Nutrition Counseling

**2. Government Schemes**

* **Rural Development**
  + Gram Yojana
  + Pradhan Mantri Awas Yojana
* **Agriculture Support**
  + Krishak Samman Yojana
  + Subsidy for Fertilizers
* **Women & Children**
  + Balika Shiksha Yojana
  + Widow Pension Yojana

**3. Education & Skill Training**

* **Vocational Courses**
  + Tailoring Classes
  + Welding Course
  + Computer Literacy
* **Scholarship & Aids**
  + SC/ST/OBC Scholarship Application
  + Free Textbook Program

**💾 Backend (Django) – Suggested Data Models**

python

CopyEdit

# models.py

class ServiceCategory(models.Model):

name = models.CharField(max\_length=100)

class ServiceSubCategory(models.Model):

name = models.CharField(max\_length=100)

category = models.ForeignKey(ServiceCategory, on\_delete=models.CASCADE)

class Service(models.Model):

name = models.CharField(max\_length=100)

description = models.TextField()

documents\_required = models.JSONField(default=list)

subcategory = models.ForeignKey(ServiceSubCategory, on\_delete=models.CASCADE)

available = models.BooleanField(default=True)

You can seed categories like:

* "Health Care"
* "Schemes"
* "Education"

Subcategories like:

* "Disease"
* "Women & Child"
* "Vocational"

**🔀 API Structure (Django REST Framework)**

| **Endpoint** | **Description** |
| --- | --- |
| /api/categories/ | List all service categories |
| /api/subcategories/?cat=Health | List subcategories under a category |
| /api/services/?subcat=Disease | List services under a subcategory |
| /api/apply/ | Apply for a specific service |

**🧩 Frontend (React) UI Structure**

**Sidebar or Tab Menu**:

css

CopyEdit

[ Health Care ]

↳ Disease

↳ Eye Test

↳ Cholera Test

[ Schemes ]

↳ Gram Yojana

↳ Balika Shiksha Yojana

[ Education ]

↳ Vocational

↳ Welding

↳ Tailoring

When a subcategory or service is clicked, show the **form to apply**, including:

* Upload documents
* Fill details (name, age, etc.)
* Submit to backend

**🧪 Example Service (React display format)**

json

CopyEdit

{

"category": "Health Care",

"subcategory": "Disease",

"service": "Eye Test",

"description": "Free vision screening for citizens",

"documents\_required": ["Aadhaar Card", "Health ID"]

}

**✅ Benefits**

* Easy to **scale** and add more services in future
* User-friendly **grouping/navigation**
* Admin can manage all services easily from a dashboard

Would you like:

* ✅ JSON data for these services to seed your database?
* ✅ A React UI component template for displaying and applying for these grouped services?

Let me know and I’ll provide you ready-to-use code.

Perfect! Here's an **extended and categorized list** of services you can include in your **Digital E Gram Panchayat** project. These services are grouped into relevant sectors like Health Care, Government Schemes, and Education, as you mentioned.

**🏥 1. Health Care Services**

These are health-related services provided to citizens, often in collaboration with primary health centers.

| **Service Name** | **Description** |
| --- | --- |
| **Eye Test Camp Registration** | Citizens can register for a free eye test at the local health center. |
| **Cholera Test Request** | Test for cholera symptoms through a health officer. |
| **General Health Checkup Camp** | Apply to attend a village-level medical camp. |
| **Vaccination Drive Enrollment** | Enroll for COVID, polio, or seasonal vaccinations. |
| **Pregnancy Health Card Request** | Register for prenatal health checkups and maternal benefits. |
| **Sanitation Inspection Request** | Request health department to inspect water or sanitation concerns. |

**🏛️ 2. Government Schemes (Yojanas)**

Citizens can apply or enroll in state/central schemes through the Gram Panchayat portal.

| **Scheme Name** | **Description** |
| --- | --- |
| **Gram Yojana Enrollment** | Apply for rural development benefits like road, drainage, and infrastructure work. |
| **Krishak Yojana** | Agricultural subsidy or equipment aid scheme for registered farmers. |
| **Balika Shiksha Yojana** | Girl child education support; includes fee waivers or cash incentives. |
| **Ujjwala Gas Yojana** | Free/subsidized gas connection for eligible rural households. |
| **Old Age Pension Yojana** | Financial support scheme for senior citizens. |
| **Widow Pension Yojana** | Financial aid for widowed women in the village. |
| **Jan Dhan Account Application** | Assist villagers in creating zero-balance bank accounts. |

**🎓 3. Education & Skill Development Services**

To help villagers, especially youth and women, gain skills for employment or self-employment.

| **Course/Service** | **Description** |
| --- | --- |
| **Vocational Training Enrollment** | Register for short-term skill training (general) |
| **Welding Training Course** | 4–6 week welding course for rural youth |
| **Tailoring Classes for Women** | Learn tailoring and stitching for self-employment |
| **Computer Literacy Program** | Basic computer usage course |
| **Digital Services Training** | Learn how to use smartphones, access government portals, etc. |
| **Youth Skill Development Scheme** | Enroll in youth-targeted job preparation programs |

**🧰 Additional Categories (Optional)**

**🚧 Infrastructure**

* Road Repair Request
* Street Light Installation
* Drainage Complaint

**🌾 Agriculture**

* Soil Health Test Request
* Seed Subsidy Application
* Water Pump Installation Request

**💡 Utilities**

* Electricity Complaint/Connection
* Drinking Water Connection Request
* Waste Management Service Request

**🛠️ How to Structure These in Your System**

**✅ Sample Service Model (Django)**

python

CopyEdit

class Service(models.Model):

CATEGORY\_CHOICES = [

('health', 'Health Care'),

('scheme', 'Government Scheme'),

('education', 'Education & Skill'),

('agriculture', 'Agriculture'),

('infrastructure', 'Infrastructure'),

('utilities', 'Utilities'),

]

title = models.CharField(max\_length=100)

description = models.TextField()

category = models.CharField(max\_length=30, choices=CATEGORY\_CHOICES)

required\_documents = models.TextField()

created\_at = models.DateTimeField(auto\_now\_add=True)

is\_active = models.BooleanField(default=True)

**📲 Example in React UI**

* **Dropdown Filter**: By category (Health, Scheme, Education...)
* **Cards/List**: Each card shows service name, description, and an "Apply" button.
* **Form Modal**: Opens application form with pre-filled service info and file upload (for documents).

Would you like:

* A **dummy JSON file** of these services for seeding your database?
* A **sample React component** to display and apply for these services?

Let me know and I can generate both for you!

**Health Care Services**

| **Service Name** | **Detailed Description** |
| --- | --- |
| **Eye Test Camp Registration** | Citizens can register online for free eye examinations conducted by qualified doctors at the local health center. These camps help identify common vision issues and provide free glasses or recommendations. |
| **Cholera Test Request** | This service allows users to request cholera screening if experiencing symptoms like diarrhea, dehydration, or stomach cramps. Health officers are dispatched to conduct home visits or refer citizens to clinics. |
| **General Health Checkup Camp** | Apply to attend periodic health checkup camps organized in the village, covering BP, sugar, anemia, and general wellness. Open to all citizens on a first-come, first-served basis. |
| **Vaccination Drive Enrollment** | Citizens can pre-register for upcoming vaccination drives (COVID-19, polio, hepatitis, etc.). Notifications will be sent via SMS or email for appointment details. |
| **Pregnancy Health Card Request** | Expecting mothers can register to receive a maternity card for accessing prenatal checkups, nutrition kits, and financial aid under maternal health programs. |
| **Sanitation Inspection Request** | Request an inspection of your locality's sanitation, water quality, or drainage system if health hazards are observed. The health department will schedule a site visit. |

**🏛️ 2. Government Schemes (Yojanas)**

| **Scheme Name** | **Detailed Description** |
| --- | --- |
| **Gram Yojana Enrollment** | Apply for infrastructure support in your locality, such as road repairs, drainage construction, or public toilets. Applications are reviewed by the gram panchayat officer. |
| **Krishak Yojana** | Farmers can apply for subsidies on seeds, fertilizers, or modern farming equipment. Scheme requires proof of land ownership or registration in the Kisan database. |
| **Balika Shiksha Yojana** | Families with school-going girls can apply for education aid. Benefits include tuition fee waivers, school kits, or monthly cash incentives for attendance. |
| **Ujjwala Gas Yojana** | Eligible women can request a new LPG gas connection at subsidized or zero cost. Requires Aadhaar verification and ration card. |
| **Old Age Pension Yojana** | Senior citizens (age 60+) can apply for monthly pension support. Requires age and income verification. Funds are directly transferred to the applicant’s bank account. |
| **Widow Pension Yojana** | Widowed women with no steady income can apply for monthly financial assistance through this welfare scheme. |
| **Jan Dhan Account Application** | This service helps villagers apply for a no-frills zero-balance savings account under the Pradhan Mantri Jan Dhan Yojana. Includes a RuPay debit card and accidental insurance. |

**🎓 3. Education & Skill Development Services**

| **Service Name** | **Detailed Description** |
| --- | --- |
| **Vocational Training Enrollment** | A general enrollment form to join certified short-term training courses in different trades based on availability. Includes orientation and placement assistance. |
| **Welding Training Course** | A 4–6 week course aimed at unemployed youth for learning welding and fabrication skills, useful for construction and manufacturing jobs. |
| **Tailoring Classes for Women** | Specially designed program for women to learn tailoring, embroidery, and sewing. Certification and starter kits are provided on completion. |
| **Computer Literacy Program** | Basic digital skills training including MS Office, email use, browsing, and accessing government services online. Targeted at students and job seekers. |
| **Digital Services Training** | Hands-on training for using smartphones, paying bills online, and accessing public services like ration card, pension portal, or health IDs. |
| **Youth Skill Development Scheme** | Prepares village youth for government or private sector jobs with mock interviews, aptitude training, and resume building workshops. |

**🧰 4. Optional Additional Categories**

**🚧 Infrastructure**

| **Service** | **Description** |
| --- | --- |
| **Road Repair Request** | Citizens can submit complaints about damaged or unpaved roads. Photos and GPS locations can be uploaded for better tracking. |
| **Street Light Installation** | Request installation of new street lights in dark or unsafe areas. Requires location details and ward member approval. |
| **Drainage Complaint** | Lodge a complaint about clogged or broken drainage lines. The sanitation team will visit and take corrective measures. |

**🌾 Agriculture**

| **Service** | **Description** |
| --- | --- |
| **Soil Health Test Request** | Farmers can request a lab test for their soil to determine nutrient levels and get personalized fertilizer suggestions. |
| **Seed Subsidy Application** | Apply for high-yield seed subsidies during sowing season. Requires farmer registration number and land records. |
| **Water Pump Installation Request** | Request support for installing an irrigation water pump, either through government subsidy or village-level funds. |

**💡 Utilities**

| **Service** | **Description** |
| --- | --- |
| **Electricity Complaint/Connection** | Apply for a new electricity connection or report outages and faults in your area. Staff will respond with service visit time. |
| **Drinking Water Connection Request** | Citizens can apply for piped water supply to their household under rural water schemes. |
| **Waste Management Service Request** | Request waste collection bins, pickup services, or awareness programs in your area. Useful for cleanliness drives and Swachh Bharat reporting. |

I want user apply form details using upper service name where user can apply

**ChatGPT said:**

Updated saved memory

Great! Here's a plan and detailed structure for building the **User Application Form** in your Digital E Gram Panchayat project, where users can apply for services like Eye Test Camp, Gram Yojana, Vocational Training, etc.

**✅ Key Features of the User Application Form**

**1. Form Fields (Common for All Services)**

These fields will appear for every service application:

| **Field Name** | **Type** | **Description** |
| --- | --- | --- |
| Full Name | Text | Applicant's full name |
| Aadhaar Number | Text (12 digits) | For identity verification |
| Mobile Number | Text | Contact number (OTP optional) |
| Email (optional) | Email | Optional email for notifications |
| Address | Textarea | Village/locality details |
| Category of Service | Dropdown | Health, Scheme, Education, Infrastructure… |
| Specific Service Name | Dropdown | Auto-populated based on category |
| Purpose/Notes | Textarea | Reason or additional info |
| Upload Document | File Upload | (Optional) e.g., Aadhaar, land proof, etc. |

**🧠 Frontend (React)**

**✅ ServiceApplyForm.jsx**

jsx

CopyEdit

import React, { useState } from 'react';

import axios from 'axios';

const categories = {

Health: [

"Eye Test Camp Registration",

"Cholera Test Request",

"General Health Checkup Camp",

"Vaccination Drive Enrollment",

"Pregnancy Health Card Request",

"Sanitation Inspection Request",

],

Scheme: [

"Gram Yojana Enrollment",

"Krishak Yojana",

"Balika Shiksha Yojana",

"Ujjwala Gas Yojana",

"Old Age Pension Yojana",

"Widow Pension Yojana",

"Jan Dhan Account Application"

],

Education: [

"Vocational Training Enrollment",

"Welding Training Course",

"Tailoring Classes for Women",

"Computer Literacy Program",

"Digital Services Training",

"Youth Skill Development Scheme"

],

// Add Infrastructure, Agriculture, Utilities as needed

};

export default function ServiceApplyForm() {

const [formData, setFormData] = useState({

fullName: '',

aadhaar: '',

mobile: '',

email: '',

address: '',

category: '',

service: '',

purpose: '',

document: null

});

const handleChange = (e) => {

const { name, value, files } = e.target;

setFormData({

...formData,

[name]: files ? files[0] : value

});

};

const handleSubmit = async (e) => {

e.preventDefault();

const data = new FormData();

for (const key in formData) {

data.append(key, formData[key]);

}

try {

const res = await axios.post('/api/applications', data);

alert("Application submitted successfully!");

} catch (err) {

alert("Error submitting application.");

}

};

return (

<form onSubmit={handleSubmit} className="p-4 max-w-xl mx-auto bg-white rounded shadow">

<h2 className="text-xl font-bold mb-4">Apply for a Service</h2>

<input name="fullName" type="text" placeholder="Full Name" className="input" onChange={handleChange} required />

<input name="aadhaar" type="text" placeholder="Aadhaar Number" className="input" onChange={handleChange} required />

<input name="mobile" type="text" placeholder="Mobile Number" className="input" onChange={handleChange} required />

<input name="email" type="email" placeholder="Email (optional)" className="input" onChange={handleChange} />

<textarea name="address" placeholder="Full Address" className="textarea" onChange={handleChange} required></textarea>

<select name="category" className="select" onChange={handleChange} required>

<option value="">Select Category</option>

{Object.keys(categories).map(cat => (

<option key={cat} value={cat}>{cat}</option>

))}

</select>

{formData.category && (

<select name="service" className="select" onChange={handleChange} required>

<option value="">Select Service</option>

{categories[formData.category].map(service => (

<option key={service} value={service}>{service}</option>

))}

</select>

)}

<textarea name="purpose" placeholder="Purpose / Additional Notes" className="textarea" onChange={handleChange} />

<input name="document" type="file" className="input" onChange={handleChange} />

<button type="submit" className="btn btn-primary mt-4">Submit Application</button>

</form>

);

}

**⚙️ Backend (Express + MongoDB)**

**✅ Mongoose Schema (models/Application.js)**

js

CopyEdit

const mongoose = require('mongoose');

const applicationSchema = new mongoose.Schema({

fullName: String,

aadhaar: String,

mobile: String,

email: String,

address: String,

category: String,

service: String,

purpose: String,

documentPath: String,

status: { type: String, default: "Pending" },

appliedAt: { type: Date, default: Date.now }

});

module.exports = mongoose.model("Application", applicationSchema);

**✅ Express Route (routes/application.js)**

js

CopyEdit

const express = require('express');

const multer = require('multer');

const Application = require('../models/Application');

const router = express.Router();

const upload = multer({ dest: 'uploads/' });

router.post('/', upload.single('document'), async (req, res) => {

try {

const data = {

...req.body,

documentPath: req.file?.path

};

const newApp = new Application(data);

await newApp.save();

res.status(201).json({ message: "Application submitted" });

} catch (err) {

res.status(500).json({ error: "Submission failed" });

}

});

module.exports = router;

**🔗 MongoDB Example Document**

json

CopyEdit

{

"fullName": "Ravi Kumar",

"aadhaar": "123456789012",

"mobile": "9876543210",

"email": "ravi@example.com",

"address": "Ward 5, Madhavpur",

"category": "Health",

"service": "Eye Test Camp Registration",

"purpose": "Blurry vision in both eyes",

"documentPath": "uploads/aadhaar.jpg",

"status": "Pending",

"appliedAt": "2025-06-06T12:34:56.789Z"

}

<https://lsgkerala.gov.in/system/files/inline-images/global-expo-conclave-2023.png>

<https://secure.mygov.in/sites/all/themes/mygov/images/panchayatiraj2022/panchayati-raj-banner.jpg>

<https://www.studyiq.com/articles/wp-content/uploads/2025/04/23184249/1200-x-628-70.jpg>

<https://cdnbbsr.s3waas.gov.in/s3e6c2dc3dee4a51dcec3a876aa2339a78/uploads/2023/07/2023072143.jpg>

<https://cdnbbsr.s3waas.gov.in/s3e6c2dc3dee4a51dcec3a876aa2339a78/uploads/2024/12/2024123094124572.jpeg>

<https://cdnbbsr.s3waas.gov.in/s3e6c2dc3dee4a51dcec3a876aa2339a78/uploads/2025/02/202502061433080426.jpeg>

<https://cdnbbsr.s3waas.gov.in/s3e6c2dc3dee4a51dcec3a876aa2339a78/uploads/2025/02/20250206984405454.jpg>

1Ab!1Ab!

- 1234567809

1236958470