EDUCATIONAL ORGANISATION USING SERVICE NOW

Submitted by S. Sneha Vyshnavi

TABLE OF CONTENTS:

Project Documentation: Educational Management System on ServiceNow

1. Introduction

This document outlines the project plan for developing and implementing an Educational Management System (EMS) on the ServiceNow platform for educational organizations. The primary goal of this system is to streamline administrative tasks, improve efficiency, and enhance the overall management of educational institutions by providing a unified, accessible, and automated platform for critical operations.

2. Project Objectives

The key objectives of this project are to:

- Centralize Data Management: Establish a single, secure, and accurate repository for all student, teacher, course, and administrative data, accessible to authorized personnel.
- Automate Core Processes: Significantly reduce manual effort and human error by automating workflows for student admissions, teacher assignments, and academic tracking.
- Enhance Operational Efficiency: Improve response times for queries, reduce paperwork, and free up administrative staff to focus on strategic initiatives rather than routine tasks.
- Improve Student & Teacher Experience: Provide self-service portals where students can access their academic records, apply for courses, and teachers can manage their rosters and submit grades efficiently.
- Enable Data-Driven Decision Making: Offer robust reporting and analytics capabilities to provide actionable insights into student performance, enrollment trends, resource utilization, and operational bottlenecks.
- Ensure Compliance & Security: Implement strict security measures and adherence to relevant data privacy regulations (e.g., FERPA, GDPR, local education board guidelines) to protect sensitive information.

3. Scope of the Project

The initial phase of this project will focus on the following core modules and their integration:

- **Student Management:**
    - Student Profile: Comprehensive capture and management of student personal information, contact details, emergency contacts, medical history, enrollment details, academic history, and disciplinary records.
    - Enrollment & Registration: Management of course registration, class assignments, and withdrawal processes.

- Reporting: Generation of customizable reports on student demographics, attendance, and performance.
- *Image Placeholder: Screenshot of a proposed student profile page in ServiceNow.*

- **Teacher Management:**
  - Teacher Profile: Centralized repository for teacher personal details, qualifications, certifications, employment history, contact information, and assigned courses/classes.
  - Workload Management: Tools for assigning courses, managing class schedules, and tracking teacher availability.
  - Performance Tracking: Basic features for recording teacher evaluations and professional development activities.
  - *Image Placeholder: Mockup of a teacher directory within the ServiceNow EMS showing assigned courses.*
- **Admission Management:**
  - Online Application Portal: User-friendly portal for prospective students to submit applications, upload required documents, and track application status.
  - Automated Workflow: Configurable workflows for application review, verification, interview scheduling, and decision-making by admissions committees.
  - Offer & Enrollment Confirmation: Automated generation and delivery of offer letters, and tracking of enrollment confirmations.
  - Analytics: Insights into application volume, conversion rates, and demographic breakdown of applicants.
  - *Image Placeholder: Wireframe of an online admission application form integrated with ServiceNow's workflow engine.*
- **Student Progress Monitoring (Academic Records):**
  - Grade Management: Secure system for teachers to submit and manage grades for assignments, quizzes, and exams.
  - Attendance Tracking: Automated or manual recording of student attendance for classes and other school activities.
  - Progress Reports: Generation of customizable progress reports and report cards for students and parents, accessible via a portal.

    - Alerts & Notifications: Automated alerts for failing grades, excessive absences, or other academic concerns to relevant staff and parents.
    - *Image Placeholder: Example of a student progress report dashboard in ServiceNow, showing grades and attendance summary.*
- **Communication Portal (Basic):**
    - Announcements: Ability for administrators and teachers to post important announcements.
    - Secure Messaging: Basic internal messaging system for authorized users (e.g., teacher-to-administrator).

Potential Future Phases (Out of Initial Scope):

- Curriculum & Course Management: Tools for managing course catalogs, curriculum outlines, and learning resources.
- Fee Management & Billing: Integration with financial systems for tuition fee collection, invoice generation, and payment tracking.
- Timetable & Scheduling: Advanced scheduling tools for classes, exams, and school events.
- Alumni Management: Database and engagement tools for former students.
- Library Management System Integration: Interfacing with existing library systems.

4. Skills Required

The successful execution of this project demands a blend of technical and domain-specific expertise:

- **ServiceNow Development & Administration:**
    - Core ServiceNow Platform Knowledge: Deep understanding of ServiceNow architecture, tables, forms, workflows, and security models.
    - Application Development: Proficiency in building custom applications within ServiceNow Studio using App Engine.
    - Workflow & Flow Designer: Expertise in designing and implementing complex automated workflows and approvals.
    - Integration Hub: Knowledge of integrating ServiceNow with external systems (e.g., student information systems, HR systems, payment gateways) using various protocols (REST, SOAP).

- ServiceNow Service Portal: Experience in developing and customizing user-friendly self-service portals using widgets and pages.
- Reporting & Analytics: Ability to create insightful reports, dashboards, and performance analytics indicators.

- **Web Development (for Portal Customization & Integrations):**
  - HTML5: Structuring the content for web pages, especially for custom Service Portal widgets.
  - CSS3: Styling the user interface to ensure a consistent and appealing look and feel aligned with the institution's branding. This includes responsive design for various devices.
  - JavaScript (Client & Server-Side):
    - *Client-Side (AngularJS/Service Portal):* Enhancing user interaction, form validation, and dynamic content loading within the Service Portal.
    - *Server-Side (Glide Scripting):* Implementing business logic, data manipulation, and integrations within ServiceNow itself.
- Database Concepts: Understanding of relational databases and data modeling to design efficient and scalable data structures within ServiceNow.
- Business Analysis: Ability to gather, analyze, and document detailed requirements from various stakeholders within the educational institution.
- Project Management: Skills in planning, executing, and closing projects, managing timelines, resources, and risks.
- Domain Knowledge: Familiarity with educational administration processes, academic cycles, and regulatory requirements (e.g., student privacy laws).

5. Project Phases (Detailed)

1. **Phase 1: Discovery & Planning**
   - Activities: Stakeholder interviews, detailed requirements gathering, current process mapping ("as-is" analysis), definition of "to-be" processes, gap analysis, detailed scope definition, project plan finalization, resource allocation, risk assessment.
   - Deliverables: Detailed Requirements Document (DRD), Project Plan, High-Level Solution Design, Risk Register.
2. **Phase 2: Design & Development**

- Activities: ServiceNow application design (data models, UI/UX mockups), workflow design, integration specifications, coding and configuration of modules, custom widget development for Service Portal, data migration strategy development.
- Deliverables: Detailed Solution Design, Developed ServiceNow Modules (Student, Teacher, Admission, Progress), Custom Service Portal Components, Integration Specifications.

3. **Phase 3: Testing & Quality Assurance**
    - Activities: Unit testing, integration testing, user acceptance testing (UAT) with key stakeholders, performance testing, security testing, bug fixing and re-testing.
    - Deliverables: Test Plans, Test Cases, UAT Sign-off, Defect Log, Performance Test Reports, Security Audit Report.
4. **Phase 4: Deployment & Implementation**
    - Activities: Configuration of production environment, data migration from legacy systems, cut-over planning, system go-live.
    - Deliverables: Deployed EMS on Production ServiceNow Instance, Data Migration Reports, Go-Live Checklist.
5. **Phase 5: Training & User Adoption**
    - Activities: Development of training materials (user manuals, quick guides), conducting training sessions for administrators, teachers, and other end-users, establishing a support model.
    - Deliverables: Training Modules, User Manuals, Training Completion Reports, Help Desk Procedures.
6. **Phase 6: Post-Implementation Support & Maintenance**
    - Activities: Ongoing technical support, system monitoring, performance tuning, bug fixes, minor enhancements, regular system upgrades (ServiceNow releases), and continuous improvement based on user feedback.
    - Deliverables: SLA-based Support, System Performance Reports, Enhancement Backlog.

6. Technology Stack

- Platform: ServiceNow (SaaS)
- Front-end Development: HTML5, CSS3, JavaScript (AngularJS for Service Portal)

- Server-side Scripting: JavaScript (GlideRecord, GlideAjax, Business Rules, Script Includes within ServiceNow)
- Integration: REST API, SOAP, IntegrationHub (where applicable)

7. Security and Compliance

Given the sensitive nature of student and teacher data, robust security and compliance measures are paramount:

- Role-Based Access Control (RBAC): Implementing strict access controls to ensure users can only view and modify data relevant to their roles (e.g., teachers see their students' grades, not all students' grades; administrators have full access).
- Data Encryption: Ensuring data is encrypted both in transit and at rest within the ServiceNow platform.
- Audit Trails: Maintaining comprehensive audit logs of all system activities, including data access, modifications, and deletions.
- Compliance Adherence: Ensuring the system adheres to relevant educational data privacy regulations such as FERPA (Family Educational Rights and Privacy Act) in the US, GDPR (General Data Protection Regulation) in Europe, and specific local educational board guidelines (e.g., for schools in Amalapuram, Andhra Pradesh, India, adhering to Indian data protection laws).
- Regular Security Audits: Conducting periodic security assessments and penetration testing.

8. Reporting and Analytics

The EMS will provide powerful reporting and analytics capabilities through ServiceNow's built-in tools:

- Customizable Dashboards: Personalized dashboards for administrators, teachers, and department heads to view key performance indicators (KPIs) and operational metrics at a glance.
- Standard Reports: Pre-built reports for common queries (e.g., student enrollment by year, teacher workload, application status breakdowns).
- Ad-Hoc Reporting: Ability for authorized users to create custom reports on demand using filters and aggregations.
- Performance Analytics (Optional, if licensed): Advanced analytics capabilities for trend analysis, forecasting, and deeper insights into operational efficiency and student outcomes.

9. User Roles and Permissions

The system will define distinct user roles with specific permissions:

- System Administrator: Full control over system configuration, user management, and data.
- Admissions Officer: Manages admission applications, workflows, and communication with applicants.
- Registrar/Enrollment Officer: Manages student enrollment, course registration, and student records.
- Teacher: Manages their assigned classes, submits grades, tracks attendance for their students, and accesses relevant student profiles.
- Department Head: Oversees teachers within their department, manages course assignments, and views departmental performance.
- Student (via Portal): Views personal profile, academic progress (grades, attendance), registered courses, and school announcements.
- Parent/Guardian (via Portal): Views their child's academic progress, attendance, and receives school announcements.

10. Change Management and Training

Successful adoption of the new EMS hinges on effective change management and comprehensive training:

- Change Champions: Identifying and empowering key individuals within the organization to advocate for and support the new system.
- Communication Plan: Regular communication with all stakeholders about project progress, benefits, and upcoming changes.
- Tailored Training: Providing role-specific training sessions (e.g., different training for admissions vs. teachers) to ensure users are comfortable and proficient with their specific functionalities.
- User Support: Establishing clear channels for users to get assistance and report issues post-implementation (e.g., an internal IT help desk leveraging ServiceNow ITSM).

11. Expected Benefits (Quantified & Qualitative)

- Reduced Administrative Overheads: Estimated 20-30% reduction in manual data entry and processing time for admissions and record keeping.

- Faster Admission Cycle: Potentially reducing admission processing time by 40-50% due to automation.
- Improved Data Accuracy: Minimize errors associated with manual data entry, leading to more reliable reports and decisions.
- Enhanced Reporting: Provide real-time data insights, enabling proactive interventions for at-risk students and better resource allocation.
- Increased Stakeholder Satisfaction: Improved experience for students, teachers, and parents through self-service options and transparent processes.
- Scalability: The ServiceNow platform allows for easy scaling to accommodate future growth in student numbers or expansion of educational programs.
- Cost Savings: Long-term reduction in operational costs associated with paper-based processes and inefficient workflows.

12. Challenges and Mitigation Strategies

- **Data Migration Complexity:**
    - Challenge: Migrating large volumes of historical student and teacher data from disparate legacy systems can be complex and error-prone.
    - Mitigation: Develop a detailed data migration plan, perform thorough data cleansing and validation, utilize ServiceNow's data import tools, and conduct pilot migrations.
- **User Adoption Resistance:**
    - Challenge: Users may resist adopting a new system due to unfamiliarity or perceived complexity.
    - Mitigation: Involve key users early in the design phase, provide extensive and role-based training, highlight benefits, offer accessible support channels, and communicate the "why" behind the change.
- **Integration with Existing Systems:**
    - Challenge: Integrating the EMS with other existing systems (e.g., finance, HR, LMS) might present technical hurdles.
    - Mitigation: Conduct a comprehensive integration analysis, utilize ServiceNow IntegrationHub capabilities, prioritize critical integrations for the initial phase, and leverage robust APIs.
- **Security and Privacy Concerns:**

- Challenge: Ensuring absolute data security and compliance with various privacy regulations.
- Mitigation: Implement strict access controls, conduct regular security audits, adhere to ServiceNow's built-in security features, and stay updated on relevant data protection laws (e.g., India's Digital Personal Data Protection Act, 2023).

13. Conclusion

The Educational Management System built on ServiceNow represents a strategic investment for educational organizations aiming to modernize their operations, enhance efficiency, and provide a superior experience for their students and staff. By leveraging ServiceNow's powerful platform capabilities combined with expert development, this project will deliver a robust, scalable, and secure solution that meets the evolving needs of contemporary education.

EXECUTABLE CODE

FORM-1:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>New Student Record</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background: #f6f7fa;
      margin: 0;
      padding: 0;
    }
    .form-wrapper {
      max-width: 1200px;
      margin: 40px auto;
      background: #fff;
      border-radius: 8px;
      box-shadow: 0 2px 8px rgba(0,0,0,0.07);
      padding: 40px 30px 30px 30px;
    }
    h2 {
      text-align: center;
      margin-bottom: 30px;
```

```
}
form {
  display: flex;
  gap: 60px;
  justify-content: center;
}
.form-section {
  flex: 1;
  min-width: 340px;
}
.form-group {
  display: flex;
  align-items: center;
  margin-bottom: 18px;
}
.form-group label {
  flex: 0 0 140px;
  margin-right: 12px;
  font-weight: 500;
  color: #3a3a3a;
}
.form-group input,
.form-group select {
  flex: 1;
  padding: 7px 10px;
  border: 1px solid #bfc6ce;
  border-radius: 4px;
```

```
  font-size: 15px;
}
.form-actions {
  text-align: center;
  margin-top: 24px;
}
button[type="submit"] {
  padding: 7px 30px;
  font-size: 15px;
  background: #7267ef;
  color: #fff;
  border: 1px solid #7267ef;
  border-radius: 4px;
  cursor: pointer;
  transition: background 0.2s;
}
button[type="submit"]:hover {
  background: #5a50c7;
}
@media (max-width: 900px) {
  form {
    flex-direction: column;
    gap: 0;
  }
  .form-section {
    min-width: unset;
  }
```

```
    }
  </style>
</head>
<body>
  <div class="form-wrapper">
    <h2>New Student Record</h2>
    <form id="studentRecordForm">
      <!-- Left section -->
      <div class="form-section">
        <div class="form-group">
          <label for="adminNumber">Admin Number</label>
          <input type="text" id="adminNumber" name="adminNumber" value="SAL0001078"
required>
        </div>
        <div class="form-group">
          <label for="adminDate">Admin Date</label>
          <input type="date" id="adminDate" name="adminDate" required>
        </div>
        <div class="form-group">
          <label for="grade">Grade</label>
          <select id="grade" name="grade" required>
            <option value="">-- None --</option>
            <option value="Grade 1">Grade 1</option>
            <option value="Grade 2">Grade 2</option>
            <option value="Grade 3">Grade 3</option>
            <option value="Grade 4">Grade 4</option>
            <option value="Grade 5">Grade 5</option>
```

```
      <!-- Add more grades as needed -->
    </select>
  </div>
  <div class="form-group">
    <label for="studentName">Student Name</label>
    <input type="text" id="studentName" name="studentName" required>
  </div>
</div>
<!-- Right section -->
<div class="form-section">
  <div class="form-group">
    <label for="fatherName">Father Name</label>
    <input type="text" id="fatherName" name="fatherName">
  </div>
  <div class="form-group">
    <label for="motherName">Mother Name</label>
    <input type="text" id="motherName" name="motherName">
  </div>
  <div class="form-group">
    <label for="motherCell">Mother Cell</label>
    <input type="tel" id="motherCell" name="motherCell" pattern="[0-9]{10,15}">
  </div>
  <div class="form-group">
    <label for="fatherCell">Father Cell</label>
    <input type="tel" id="fatherCell" name="fatherCell" pattern="[0-9]{10,15}">
  </div>
</div>
```

```
    </form>
    <div class="form-actions">
      <button type="submit" form="studentRecordForm">Submit</button>
    </div>
    <p id="result" style="text-align:center;color:green;margin-top:18px;"></p>
  </div>
  <script>
    document.getElementById('studentRecordForm').onsubmit = function(e) {
      e.preventDefault();
      // Typically, you would send form data to your backend or ServiceNow here.
      document.getElementById('result').textContent = "Record submitted successfully!";
      document.getElementById('studentRecordForm').reset();
    };
  </script>
</body>
</html>
```

FORM-2:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Admission - New Record</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background: #f6f7fa;
      margin: 0;
      padding: 0;
    }
    .progress-bar {
      display: flex;
      justify-content: center;
      margin: 30px 0 18px 0;
      gap: 0;
    }
    .progress-step {
      background: #e2e4ea;
      color: #5c5c5c;
      padding: 8px 24px;
      border-radius: 6px 6px 0 0;
      font-size: 15px;
      border-right: 2px solid #fff;
```

```
  flex: 1;
  text-align: center;
  min-width: 100px;
}
.progress-step:last-child {
  border-right: none;
}
.form-wrapper {
  max-width: 1400px;
  margin: 0 auto;
  background: #fff;
  border-radius: 8px;
  box-shadow: 0 2px 8px rgba(0,0,0,0.07);
  padding: 30px 35px 25px 35px;
}
form {
  display: flex;
  flex-direction: row;
  gap: 40px;
  justify-content: center;
}
.form-section {
  flex: 1;
  min-width: 390px;
}
.form-group {
  display: flex;
```

```
  align-items: center;
  margin-bottom: 16px;
}
.form-group label {
  flex: 0 0 145px;
  margin-right: 10px;
  font-weight: 500;
  color: #3a3a3a;
}
.form-group input,
.form-group select {
  flex: 1;
  padding: 7px 10px;
  border: 1px solid #bfc6ce;
  border-radius: 4px;
  font-size: 15px;
}
.form-group select.currency {
  width: 60px;
  margin-right: 0;
}
.form-group .fee-fields {
  display: flex;
  align-items: center;
  width: 100%;
  gap: 8px;
}
```

```
.form-group input[type="number"] {
  max-width: 110px;
}
.form-group textarea {
  flex: 1;
  padding: 8px 10px;
  border: 1px solid #bfc6ce;
  border-radius: 4px;
  font-size: 15px;
  resize: vertical;
  min-height: 36px;
}
/* Comments and status row */
.form-row {
  display: flex;
  gap: 40px;
  align-items: flex-start;
  margin-bottom: 18px;
}
.form-row .form-group {
  flex: 1;
  margin-bottom: 0;
}
/* Tabs */
.tab-container {
  margin-top: 24px;
  background: #fff;
```

```
  border-radius: 7px;
  box-shadow: 0 2px 8px rgba(0,0,0,0.04);
  width: 94%;
  margin-left: auto;
  margin-right: auto;
  border: 1px solid #d2dae7;
  padding: 0 0 16px 0;
}
.tab-header {
  display: flex;
  border-bottom: 1px solid #e3e8f0;
  background: #f9fafb;
}
.tab-header button {
  background: none;
  border: none;
  padding: 10px 18px;
  cursor: pointer;
  font-weight: 600;
  font-size: 15px;
  border-radius: 7px 7px 0 0;
  color: #7267ef;
  border-bottom: 2px solid transparent;
  margin-bottom: -1px;
  transition: border 0.2s;
}
.tab-header button.active {
```

```
  background: #e6f1e8;
  border-bottom: 2px solid #58b368;
  color: #333;
}
.tab-pane {
  display: none;
  padding: 24px 16px 0 16px;
}
.tab-pane.active {
  display: block;
}
.tab-row {
  display: flex;
  gap: 28px;
  align-items: center;
  margin-bottom: 12px;
}
.tab-row label {
  flex: 0 0 120px;
  color: #3a3a3a;
  font-weight: 500;
}
.tab-row select {
  flex: 1;
  padding: 6px 10px;
  border: 1px solid #bfc6ce;
  border-radius: 4px;
```

```
  font-size: 15px;
  background: #fff;
}
.form-actions {
  text-align: left;
  margin: 30px 0 0 100px;
}
button[type="submit"] {
  padding: 7px 30px;
  font-size: 15px;
  background: #fff;
  color: #7267ef;
  border: 1.5px solid #7267ef;
  border-radius: 4px;
  cursor: pointer;
  transition: background 0.2s, color 0.2s;
  margin-top: 8px;
}
button[type="submit"]:hover {
  background: #7267ef;
  color: #fff;
}
@media (max-width: 1000px) {
  form { flex-direction: column; gap: 0; }
  .form-section { min-width: unset; }
  .form-row { flex-direction: column; gap: 0; }
}
```

```
  </style>
</head>
<body>
  <div class="progress-bar">
    <div class="progress-step">New</div>
    <div class="progress-step">In progress</div>
    <div class="progress-step">Joined</div>
    <div class="progress-step">Rejected</div>
    <div class="progress-step">Rejoined</div>
    <div class="progress-step">Closed</div>
    <div class="progress-step">Cancelled</div>
  </div>
  <div class="form-wrapper">
    <form id="admissionForm" autocomplete="off">
      <div class="form-section">
        <div class="form-group">
          <label for="admissionNumber">Admission Number</label>
          <input type="text" id="admissionNumber" name="admissionNumber" required>
        </div>
        <div class="form-group">
          <label for="purposeOfJoin">Purpose of join</label>
          <select id="purposeOfJoin" name="purposeOfJoin" required>
            <option value="">-- None --</option>
            <option value="Admission">Admission</option>
            <option value="Transfer">Transfer</option>
            <option value="Re-admission">Re-admission</option>
            <!-- Add more options as needed -->
```

```
  </select>
 </div>
 <div class="form-group">
  <label for="studentName">Student Name</label>
  <input type="text" id="studentName" name="studentName" required>
 </div>
 <div class="form-group">
  <label for="fatherName">Father Name</label>
  <input type="text" id="fatherName" name="fatherName">
 </div>
 <div class="form-group">
  <label for="motherName">Mother Name</label>
  <input type="text" id="motherName" name="motherName">
 </div>
 <div class="form-group">
  <label for="comments">Comments</label>
  <textarea id="comments" name="comments"></textarea>
 </div>
</div>
<div class="form-section">
 <div class="form-group">
  <label for="adminDate">Admin Date</label>
  <input type="date" id="adminDate" name="adminDate" required>
 </div>
 <div class="form-group">
  <label for="grade">Grade</label>
  <select id="grade" name="grade" required>
```

```
    <option value="">-- None --</option>
    <option value="Grade 1">Grade 1</option>
    <option value="Grade 2">Grade 2</option>
    <option value="Grade 3">Grade 3</option>
    <option value="Grade 4">Grade 4</option>
    <option value="Grade 5">Grade 5</option>
    <!-- Add more grades as needed -->
  </select>
</div>
<div class="form-group">
  <label for="fee">Fee</label>
  <div class="fee-fields">
    <select class="currency" id="currency" name="currency">
      <option value="$">$</option>
      <option value="₹">₹</option>
      <option value="€">€</option>
      <option value="£">£</option>
    </select>
    <input type="number" id="fee" name="fee" min="0" step="0.01" value="0.00">
  </div>
</div>
<div class="form-group">
  <label for="fatherCell">Father Cell</label>
  <input type="tel" id="fatherCell" name="fatherCell" pattern="[0-9]{10,15}">
</div>
<div class="form-group">
  <label for="motherCell">Mother Cell</label>
```

```
      <input type="tel" id="motherCell" name="motherCell" pattern="[0-9]{10,15}">
    </div>
    <div class="form-group">
      <label for="adminStatus">Admin Status</label>
      <select id="adminStatus" name="adminStatus" required>
        <option value="">-- None --</option>
        <option value="New">New</option>
        <option value="In progress">In progress</option>
        <option value="Joined">Joined</option>
        <option value="Rejected">Rejected</option>
        <option value="Rejoined">Rejoined</option>
        <option value="Closed">Closed</option>
        <option value="Cancelled">Cancelled</option>
      </select>
    </div>
  </div>
</form>
<!-- Comments/Admin Status row -->
<div class="form-row">
  <div class="form-group" style="flex: 1;">
    <label for="comments">Comments</label>
    <textarea id="commentsRow" name="commentsRow" style="width: 100%;"></textarea>
  </div>
  <div class="form-group" style="flex: 1;">
    <label for="adminStatusRow">Admin Status</label>
    <select id="adminStatusRow" name="adminStatusRow" style="width: 100%;">
      <option value="">-- None --</option>
```

```
        <option value="New">New</option>
        <option value="In progress">In progress</option>
        <option value="Joined">Joined</option>
        <option value="Rejected">Rejected</option>
        <option value="Rejoined">Rejoined</option>
        <option value="Closed">Closed</option>
        <option value="Cancelled">Cancelled</option>
      </select>
    </div>
  </div>
  <!-- Tabbed area -->
  <div class="tab-container">
    <div class="tab-header">
      <button type="button" class="active" onclick="showTab('schoolDetails')">School Details</button>
      <button type="button" onclick="showTab('address')">Address</button>
    </div>
    <div id="schoolDetails" class="tab-pane active">
      <div class="tab-row">
        <label for="schoolArea">School Area</label>
        <select id="schoolArea" name="schoolArea">
          <option value="">-- None --</option>
          <option value="North">North</option>
          <option value="South">South</option>
          <option value="East">East</option>
          <option value="West">West</option>
          <!-- Add more as needed -->
```

```
    </select>
  </div>
  <div class="tab-row">
    <label for="school">School</label>
    <select id="school" name="school">
      <option value="">-- None --</option>
      <option value="School A">School A</option>
      <option value="School B">School B</option>
      <option value="School C">School C</option>
      <!-- Add more as needed -->
    </select>
  </div>
</div>
<div id="address" class="tab-pane">
  <div class="tab-row">
    <label for="addressLine">Address</label>
    <input type="text" id="addressLine" name="addressLine" style="width: 60%;">
  </div>
  <div class="tab-row">
    <label for="city">City</label>
    <input type="text" id="city" name="city" style="width: 60%;">
  </div>
  <div class="tab-row">
    <label for="zip">Zip</label>
    <input type="text" id="zip" name="zip" style="width: 60%;">
  </div>
</div>
```

```
    </div>
    <div class="form-actions">
      <button type="submit" form="admissionForm">Submit</button>
    </div>
    <p id="result" style="text-align:center;color:green;margin-top:18px;"></p>
  </div>
  <script>
    // Tab switching logic
    function showTab(tabId) {
      document.querySelectorAll('.tab-pane').forEach(pane => pane.classList.remove('active'));
      document.querySelectorAll('.tab-header button').forEach(btn =>
btn.classList.remove('active'));
      document.getElementById(tabId).classList.add('active');
      if(tabId === 'schoolDetails') {
        document.querySelector('.tab-header button:nth-child(1)').classList.add('active');
      } else {
        document.querySelector('.tab-header button:nth-child(2)').classList.add('active');
      }
    }
    // Form submission
    document.getElementById('admissionForm').onsubmit = function(e) {
      e.preventDefault();
      document.getElementById('result').textContent = "Admission record submitted
successfully!";
      document.getElementById('admissionForm').reset();
    };
  </script>
</body></html>
```

FORM-3:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>New Section - Student Progress</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background: #f6f7fa;
      margin: 0;
      padding: 0;
    }
    .form-wrapper {
      max-width: 1600px;
      margin: 0 auto;
      background: #fff;
      border-radius: 8px;
      box-shadow: 0 2px 8px rgba(0,0,0,0.07);
      padding: 38px 20px 18px 20px;
      min-width: 1200px;
    }
    form {
      width: 100%;
    }
    .form-row {
      display: flex;
```

```
  gap: 32px;
  margin-bottom: 8px;
  align-items: center;
}
.form-row label {
  width: 145px;
  font-weight: 500;
  color: #3a3a3a;
}
.form-row input,
.form-row select {
  flex: 1;
  padding: 7px 10px;
  border: 1px solid #bfc6ce;
  border-radius: 4px;
  font-size: 15px;
}
.form-row select {
  max-width: 220px;
}
.side-by-side {
  display: flex;
  gap: 32px;
  margin-bottom: 6px;
}
.side-by-side > .column {
  flex: 1;
```

```
}
/* Student Progress styles */
.progress-section {
  background: #f6f7fa;
  border-radius: 8px;
  margin: 40px 0 0 0;
  padding: 36px 0 32px 0;
  border-top: 1px solid #e5e6ea;
  border-bottom: 1px solid #e5e6ea;
}
.progress-title {
  font-size: 2rem;
  text-align: left;
  margin-left: 115px;
  margin-bottom: 26px;
  font-weight: 500;
  letter-spacing: -.5px;
}
.progress-fields {
  display: flex;
  gap: 40px;
  justify-content: center;
}
.progress-subjects, .progress-summary {
  flex: 1;
  max-width: 350px;
}
```

```
.progress-row {
  display: flex;
  align-items: center;
  margin-bottom: 13px;
}
.progress-row label {
  width: 90px;
  font-weight: 500;
  color: #3a3a3a;
}
.progress-row input {
  flex: 1;
  padding: 7px 10px;
  border: 1px solid #bfc6ce;
  border-radius: 4px;
  font-size: 15px;
  background: #fff;
}
.progress-row input[readonly] {
  background: #e9ecf2;
}
.submit-actions {
  margin: 30px 0 0 110px;
}
button[type="submit"] {
  padding: 7px 30px;
  font-size: 15px;
```

```
    background: #fff;
    color: #7267ef;
    border: 1.5px solid #7267ef;
    border-radius: 4px;
    cursor: pointer;
    transition: background 0.2s, color 0.2s;
    margin-top: 8px;
  }
  button[type="submit"]:hover {
    background: #7267ef;
    color: #fff;
  }
  @media (max-width: 1300px) {
    .form-wrapper { min-width: unset; }
    .progress-fields { flex-direction: column; }
    .progress-title { margin-left: 0; text-align: center; }
    .submit-actions { margin-left: 0; text-align: center; }
  }
 </style>
</head>
<body>
 <div class="form-wrapper">
  <form id="sectionProgressForm" autocomplete="off">
    <!-- Section 1: Student & parent info -->
    <div class="side-by-side">
     <div class="column">
      <div class="form-row">
```

```
    <label for="admissionNumber">Admission Number</label>
    <input type="text" id="admissionNumber" name="admissionNumber" required>
  </div>
  <div class="form-row">
    <label for="grade">Grade</label>
    <select id="grade" name="grade" required>
      <option value="">-- None --</option>
      <option value="Grade 1">Grade 1</option>
      <option value="Grade 2">Grade 2</option>
      <option value="Grade 3">Grade 3</option>
      <option value="Grade 4">Grade 4</option>
      <option value="Grade 5">Grade 5</option>
      <!-- Add more grades as needed -->
    </select>
  </div>
  <div class="form-row">
    <label for="studentName">Student Name</label>
    <input type="text" id="studentName" name="studentName" required>
  </div>
</div>
<div class="column">
  <div class="form-row">
    <label for="fatherName">Father Name</label>
    <input type="text" id="fatherName" name="fatherName">
  </div>
  <div class="form-row">
    <label for="motherName">Mother Name</label>
```

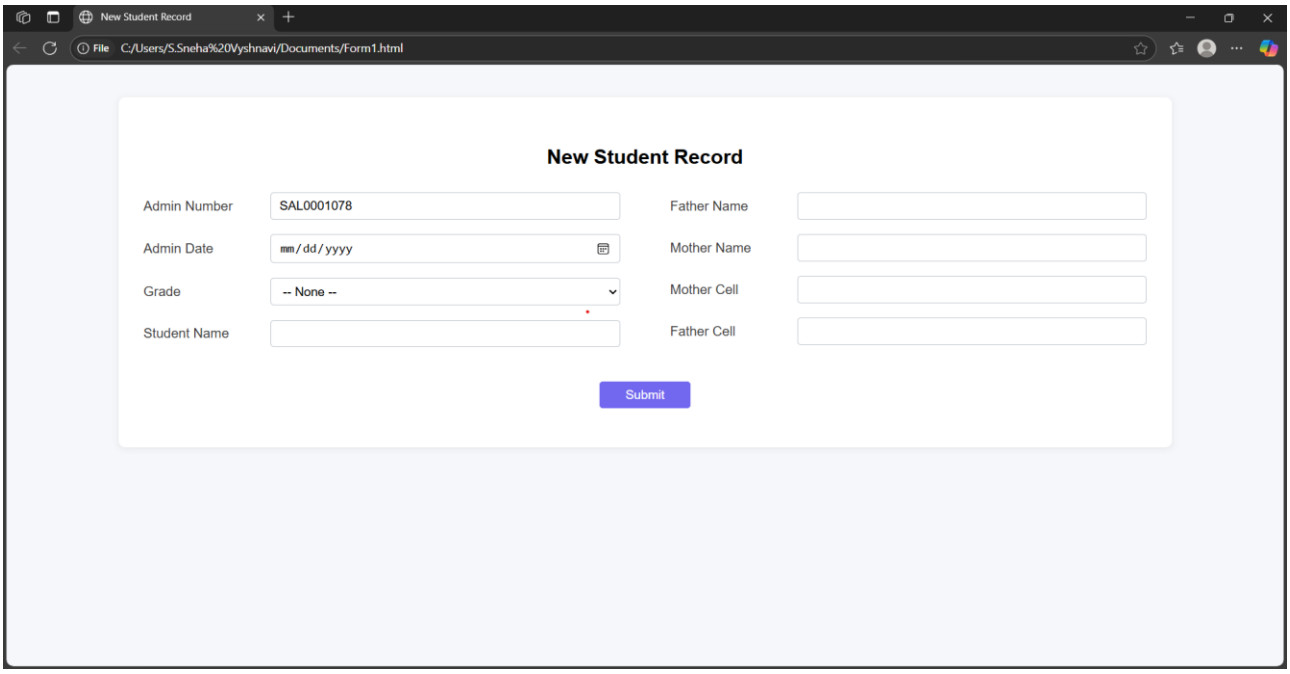
```
      <input type="text" id="motherName" name="motherName">
    </div>
    <div class="form-row">
      <label for="fatherCell">Father Cell</label>
      <input type="tel" id="fatherCell" name="fatherCell" pattern="[0-9]{10,15}">
    </div>
    <div class="form-row">
      <label for="motherCell">Mother Cell</label>
      <input type="tel" id="motherCell" name="motherCell" pattern="[0-9]{10,15}">
    </div>
  </div>
</div>
<!-- Divider -->
<div class="progress-section">
  <div class="progress-title">Student Progress</div>
  <div class="progress-fields">
    <div class="progress-subjects">
      <div class="progress-row">
        <label for="telugu">Telugu</label>
        <input type="number" id="telugu" name="telugu" min="0" max="100" step="1">
      </div>
      <div class="progress-row">
        <label for="hindi">Hindi</label>
        <input type="number" id="hindi" name="hindi" min="0" max="100" step="1">
      </div>
      <div class="progress-row">
        <label for="english">English</label>
```

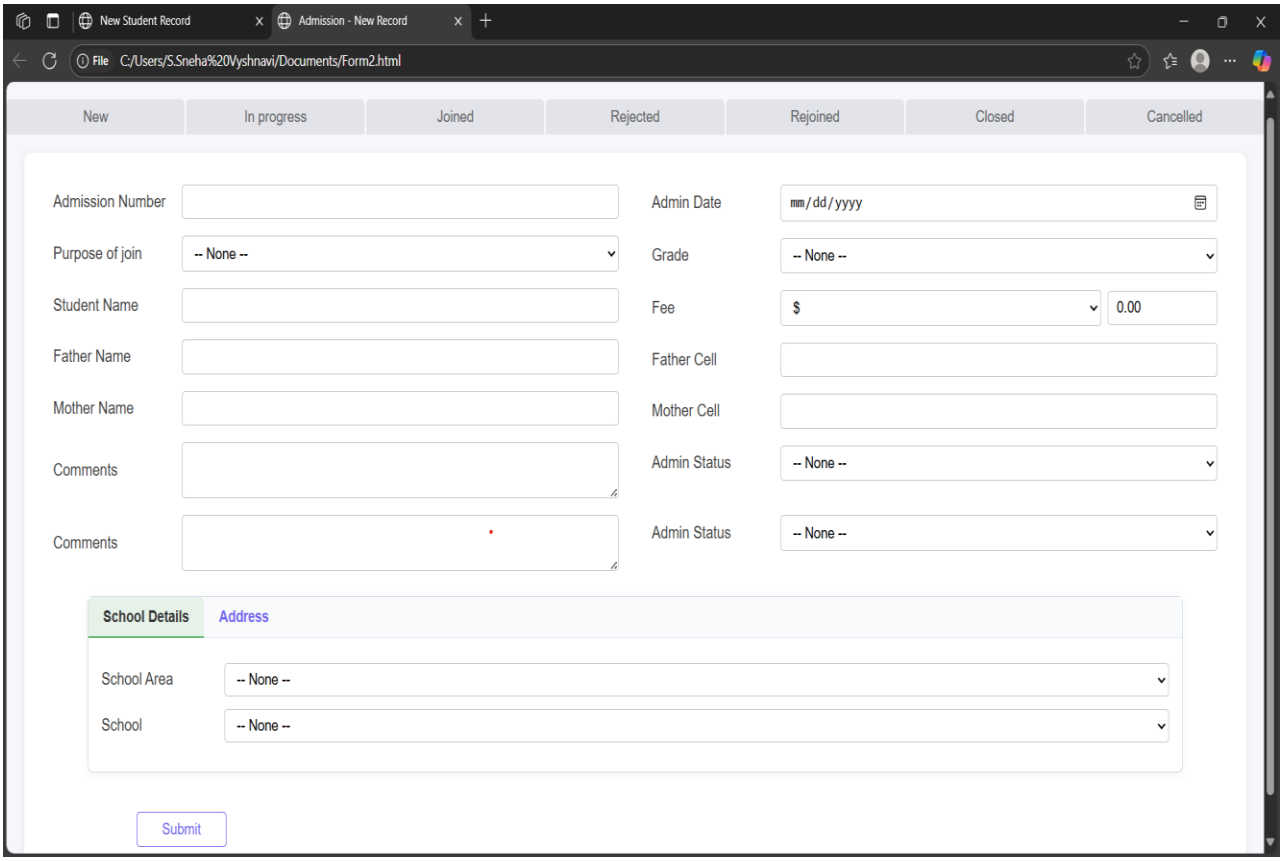
```
    <input type="number" id="english" name="english" min="0" max="100" step="1">
  </div>
  <div class="progress-row">
    <label for="maths">Maths</label>
    <input type="number" id="maths" name="maths" min="0" max="100" step="1">
  </div>
  <div class="progress-row">
    <label for="science">Science</label>
    <input type="number" id="science" name="science" min="0" max="100" step="1">
  </div>
  <div class="progress-row">
    <label for="social">Social</label>
    <input type="number" id="social" name="social" min="0" max="100" step="1">
  </div>
 </div>
 <div class="progress-summary">
  <div class="progress-row">
    <label for="total">Total</label>
    <input type="number" id="total" name="total" readonly>
  </div>
  <div class="progress-row">
    <label for="percentage">Percentage</label>
    <input type="text" id="percentage" name="percentage" readonly>
  </div>
  <div class="progress-row">
    <label for="result">Result</label>
    <input type="text" id="result" name="result" readonly>
```

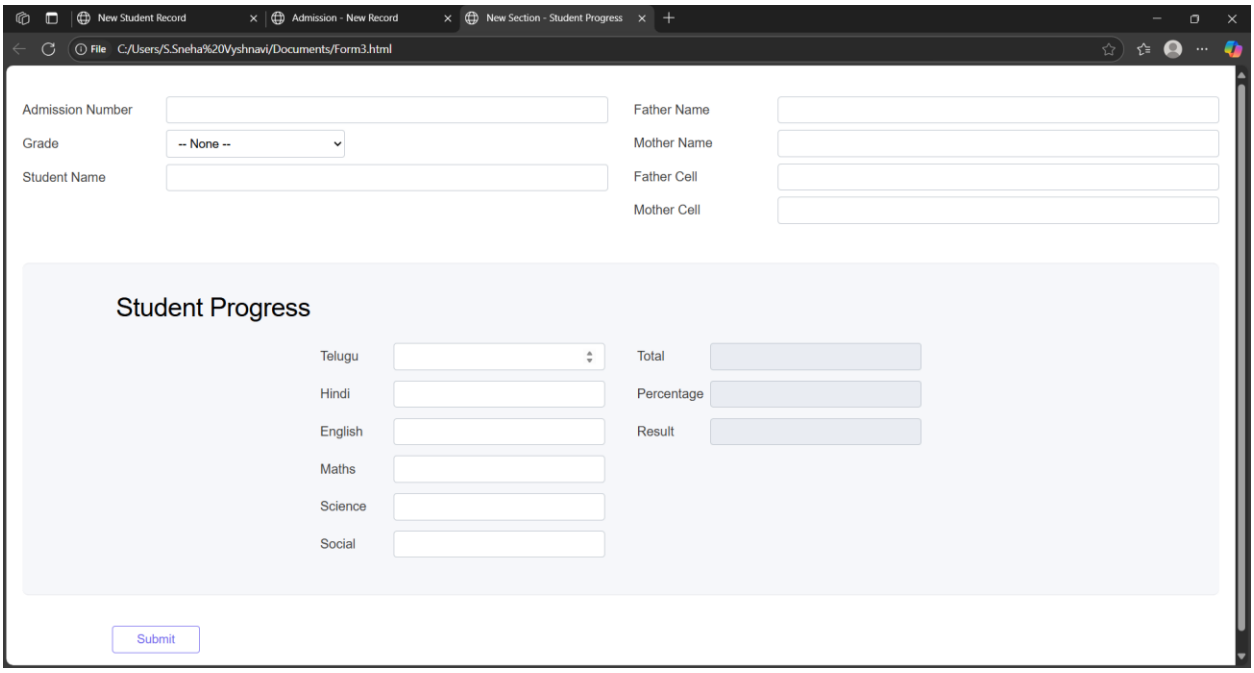
```
        </div>
       </div>
      </div>
    </div>
    <div class="submit-actions">
      <button type="submit">Submit</button>
    </div>
    <p id="formResult" style="text-align:center;color:green;margin-top:18px;"></p>
  </form>
</div>
<script>
  // Calculate total, percentage, and result on input
  function updateProgressSummary() {
    const subjects = ['telugu','hindi','english','maths','science','social'];
    let total = 0, count = 0, allValid = true, pass = true;
    for(let subj of subjects) {
      let val = parseInt(document.getElementById(subj).value, 10);
      if(!isNaN(val)) {
        total += val;
        count++;
        if(val < 35) pass = false;
      } else {
        allValid = false;
      }
    }
    document.getElementById('total').value = allValid && count === 6 ? total : '';
    let percentage = allValid && count === 6 ? (total/6).toFixed(2) : '';
```

```
    document.getElementById('percentage').value = percentage ? percentage + ' %' : '';
    document.getElementById('result').value = percentage ? (pass ? 'Pass' : 'Fail') : '';
  }
  ['telugu','hindi','english','maths','science','social'].forEach(id => {
    document.getElementById(id).addEventListener('input', updateProgressSummary);
  });
  document.getElementById('sectionProgressForm').onsubmit = function(e) {
    e.preventDefault();
    document.getElementById('formResult').textContent = "Student section and progress submitted successfully!";
    document.getElementById('sectionProgressForm').reset();
    updateProgressSummary();
  };
 </script>
</body>
</html>
```

RESULT


New Student Record
File C:/Users/S.Sneha%20Vyshnavi/Documents/Form1.html
New Student Record
Admin Number
SAL0001078
Admin Date
mm/dd/yyyy
Grade
-- None --
Student Name
Father Name
Mother Name
Mother Cell
Father Cell
Submit


New Student Record
Admission - New Record
File C:/Users/S.Sneha%20Vyshnavi/Documents/Form2.html
New
In progress
Joined
Rejected
Rejoined
Closed
Cancelled
Admission Number
Admin Date
mm/dd/yyyy
Purpose of join
-- None --
Grade
-- None --
Student Name
Fee
$
0.00
Father Name
Father Cell
Mother Name
Mother Cell
Comments
Admin Status
-- None --
Comments
Admin Status
-- None --
School Details
Address
School Area
-- None --
School
-- None --
Submit


New Student Record
Admission - New Record
New Section - Student Progress
File C:/Users/S.Sneha%20Vyshnavi/Documents/Form3.html
Admission Number
Grade
-- None --
Student Name
Father Name
Mother Name
Father Cell
Mother Cell
Student Progress
Telugu
Hindi
English
Maths
Science
Social
Total
Percentage
Result
Submit