```
import type { PlayerDevelopmentPlanOutput } from '@/ai/schemas';
import type { PlayerStats } from './data';

// Interfaces

export interface TechnicalSkills {
  batting: {
    timing: number;
    shotSelection: number;
    defense: number;
    powerHitting: number;
    footwork: number;
    playingSpin: number;
    runningBetweenWickets: number;
    concentration: number;
    patience: number;
  };
  bowling: {
    paceSpeed: number;
    accuracy: number;
    spinVariation: number;
    controlOfVariations: number;
    movement: number;
    bouncerShortBall: number;
    staminaWorkload: number;
    openingSpellImpact: number;
    deathPowerplaySkills: number;
  };
  fielding: {
    agilitySpeed: number;
    catchingClose: number;
    catchingOutfield: number;
    throwingAccuracy: number;
    throwingPower: number;
    groundFielding: number;
    reflexes: number;
    stumpingRunOuts: number;
    gatheringCollecting: number;
  };
}

export interface MentalSkills {
  composure: number;
  aggression: number;
  anticipation: number;
  determination: number;
  concentration: number;
  decisions: number;
  bravery: number;
  leadership: number;
  teamwork: number;
  workRate: number;
}

export interface PhysicalSkills {
  pace: number;
  acceleration: number;
  agility: number;
  stamina: number;
  strength: number;
  balance: number;
  naturalFitness: number;
  jumpingReach: number;
}

export interface PersonSkills {
  technical: TechnicalSkills;
  mental: MentalSkills;
  physical: PhysicalSkills;
}


export interface Person {
  personId: string;
  firstName: string;
  lastName: string;
  displayName?: string;
  dateOfBirth?: Date;
  email: string;
  phone?: string;
  emergencyContact?: {
    name: string;
    relation: string;
    phone: string;
  };
  profileImageUrl?: string;
  roles: string[];
  activeRole: string;
  assignedSchools?: string[];
  physicalAttributes?: {
    heightCm?: number;
    weightKg?: number;
    battingHand?: "Left"|"Right";
    bowlingHand?: "Left"|"Right";
    bowlingStyles?: string[];
  };
  biography?: string;
  qualifications?: string[];
  notificationPreferences?: {
    email: boolean;
    push: boolean;
  };
  userId?: string;
  developmentPlan?: PlayerDevelopmentPlanOutput;
  developmentPlanGeneratedAt?: Date;
  fcmTokens?: string[];
  skills?: PersonSkills;
  status?: 'active' | 'pending_review' | 'inactive';
  requestedSchoolName?: string;
  usedInviteCode?: string;
  requestedPlayerLink?: string;
}

export interface School {
  schoolId: string;
  name: string;
  abbreviation?: string;
  motto?: string;
  establishmentYear?: number;
  principal?: string;
  socialMedia?: {
    facebook?: string;
    twitter?: string;
    instagram?: string;
    youtube?: string;
  };
  logoUrl?: string;
  website?: string;
  phone?: string;
  location?: string;
  brandColors?: {
    primary: string;
    secondary: string;
  };
  staff?: Person[]; // No longer directly stored, but added for client components
}

export interface Division {
  divisionId: string;
  name: string;
}

export interface Season {
  seasonId: string;
  name: string;
  startDate: Date;
  endDate: Date;
  active: boolean;
}

export interface FieldAssignment {
  assignmentId: string;
  personId: string;
  personName: string;
}

export interface SurfaceCondition {
  rating: number; // 1-5
  details?: Record<string, string | number>;
}

export interface Field {
  fieldId: string;
  name: string;
  schoolId?: string;
  schoolName?: string;
  pitchType?: string;
  facilities?: string[];
  status: 'Available' | 'Maintenance' | 'Closed';
  assignments?: FieldAssignment[];
  location?: string;
  size?: string;
  amenities?: string[];
  alias?: string;
  contactPerson?: string;
  contactPhone?: string;
  notes?: string;
  surfaceCondition?: SurfaceCondition;
  coordinates?: {
    lat: number;
    lon: number;
  };
  imageUrls?: string[];
}

export interface Team {
  teamId: string;
  name: string;
  alias?: string;
  schoolId: string;
  schoolName: string;
  divisionId: string;
  divisionName: string;
  seasonId: string;
  seasonName: string;
  teamClass?: string;
  logoUrl?: string;
  teamColors?: {
    primary?: string;
    secondary?: string;
  };
  analysisReport?: string;
  abbreviation?: string;
}

export interface RosterMember {
  assignmentId: string;
  personId: string;
  personName: string;
  role: string;
  status: string;
  isCaptain: boolean;
  isViceCaptain: boolean;
}

export interface RosterMemberWithStats extends RosterMember {
    stats: PlayerStats;
    profileImageUrl?: string;
    physicalAttributes?: Person['physicalAttributes'];
}
```

```
export type MatchStatus = 'scheduled' | 'live' | 'completed' | 'postponed' | 'cancelled' |
'abandoned';
export type AvailabilityStatus = 'attending' | 'unavailable' | 'tentative' | 'injured';

export interface PlayerOfTheMatch {
  name: string;
  teamName: string;
  justification: string;
}

export interface ShotData {
  runs: number;
  angle: number; // in degrees
  distance: number; // as a ratio of radius (0 to 1)
  batsmanId: string;
}

export interface Extras {
    total: number;
    wides: number;
    noBalls: number;
    byes: number;
    legByes: number;
    partnership: number;
    partnershipStartTime?: Date;
}

export interface LiveFallOfWicket {
  wicketNumber: number;
  runs: number;
  batsmanName: string;
  timestamp: Date;
}

export type BowlingAngle = 'Over the Wicket' | 'Round the Wicket';

export interface Partnership {
  batsman1Id: string;
  batsman2Id: string;
  batsman1Name: string;
  batsman2Name: string;
  totalRuns: number;
  totalBalls: number;
  batsman1Runs: number;
  batsman1Balls: number;
  batsman2Runs: number;
  batsman2Balls: number;
}

export interface LiveScore {
    runs: number;
    wickets: number;
    overs: number;
    balls: number;
    currentOver: string[];
    ballHistory?: string[];
    onStrikeBatsmanId?: string | null;
    nonStrikerBatsmanId?: string | null;
    bowlerId?: string | null;
    lastBowlerId?: string | null;
    batsmenOut?: string[];
    liveInnings?: 1 | 2;
    shots?: ShotData[];
    batsmanStats: { [key: string]: { runs: number, balls: number, timeIn: Date | null, timeOut?: Date | null, status?: string } };
    bowlerStats: { [key: string]: { wickets: number, runsConceded: number, overs: number, balls: number, maidens: number, consecutiveWickets?: number } };
    extras: Extras;
    bowlingAngle: BowlingAngle;
    newBatsmanRequired?: boolean;
    endOfOver?: boolean;
    fallOfWickets?: LiveFallOfWicket[];
    partnerships?: Partnership[];
}

export interface Lineup {
  playingXI: string[];
  twelfthMan: string | null;
}

export interface Match {
  matchId: string;
  teamAId: string;
  teamAName: string;
  teamAAbbreviation?: string;
  teamALogoUrl?: string;
  teamBId: string;
  teamBName: string;
  teamBAbbreviation?: string;
  teamBLogoUrl?: string;
  competitionId?: string;
  competitionName?: string;
  competitionType?: string;
  seasonId?: string;
  seasonName?: string;
  divisionId?: string;
  divisionName?: string;
  fieldId: string;
  fieldName: string;
  dateTime: Date;
  status: MatchStatus;
  statusReason?: string;
  round?: number;
  report?: string;
  preview?: string;
  teamAColor?: string;
  teamBColor?: string;
  playerOfTheMatch?: PlayerOfTheMatch;
  winnerTeamId?: string;
  result?: string;
  audioCommentaryUrl?: string;
  analysisReports?: { [teamId: string]: string; };
  firstInningsTotal?: number;
  liveScore?: LiveScore;
  previousLiveScore?: LiveScore;
  firstInningsLiveScore?: LiveScore;
  userId?: string;
  availability?: { [personId: string]: { status: AvailabilityStatus; note?: string; } };
  lineupConfirmedByCaptainA: boolean;
  lineupConfirmedByCaptainB: boolean;
  tossWinnerId?: string;
  tossDecision?: 'Bat' | 'Bowl';
}

export interface Official {
  assignmentId: string;
  personId: string;
  personName: string;
  role: string;
  confirmed: boolean;
```

```
}

// Scorecard related types
export interface BatsmanStats {
  name: string;
  status: string; // e.g., "b. Bowler", "not out"
  runs: number;
  balls: number;
  fours: number;
  sixes: number;
  strikeRate: number;
  timeAtCrease?: number; // in minutes
}

export interface BowlerStats {
  name: string;
  overs: number;
  maidens: number;
  runs: number;
  wickets: number;
  economy: number;
}

export interface FallOfWicket {
    runs: number;
    wicket: number;
    batsmanName: string;
    over: number;
}

export interface Innings {
  teamName: string;
  totalRuns: number;
  wickets: number;
  overs: number;
  battingCard: BatsmanStats[];
  bowlingCard: BowlerStats[];
  fallOfWickets: FallOfWicket[];
  extras: { total: number; details: string };
}

// Stats related types
export interface PlayerStats {
    matchesPlayed: number;
    inningsBatted: number;
    notOuts: number;
    totalRuns: number;
    highestScore: number;
    highestScoreNotOut: boolean;
    ballsFaced: number;
    battingAverage: number;
    strikeRate: number;
    hundreds: number;
    fifties: number;
    fours: number;
    sixes: number;
    oversBowled: number;
    runsConceded: number;
    maidens: number;
    wicketsTaken: number;
    bowlingAverage: number;
    economyRate: number;
    bestBowling: string;
    bestBowlingWickets: number;
    bestBowlingRuns: number;
    catches: number;
    stumpings: number;
}

export interface TeamStats {
    matchesPlayed: number;
    matchesWon: number;
    matchesLost: number;
    matchesDrawn: 0;
    totalRunsScored: number;
    totalWicketsTaken: number;
    netRunRate: number;
}

export interface PlayerTeamAssignment {
  assignmentId: string;
  teamId: string;
  teamName: string;
  role: string;
  status: string;
  isCaptain: boolean;
  isViceCaptain: boolean;
}
```

```
export interface Vehicle {
  vehicleId: string;
  name: string;
  type: 'Bus' | 'Minibus' | 'Van' | 'Car';
  capacity: number;
  registration: string;
}

export interface TransportAssignment {
  assignmentId: string;
  vehicleId: string;
  vehicleName: string;
  vehicleType: Vehicle['type'];
  driverId: string;
  driverName: string;
}

export interface FullTransportAssignment extends TransportAssignment {
  matchId: string;
  matchName: string;
  dateTime: Date;
}

export interface MatchForecast {
    summary: string;
    details: {
        temperature: number;
        condition: string;
        precipitationChance: number;
        windSpeed: number;
    }
}

export interface LeaderboardPlayer extends Person {
    stats: PlayerStats;
}

export interface StandingTeam extends Team {
    stats: TeamStats;
}

export interface Competition {
  competitionId: string;
  name: string;
  type: 'League' | 'Cup' | 'Tournament' | 'Festival' | 'Friendlies';
  competitionClass?: string;
  seasonId: string;
  seasonName: string;
  divisionId: string;
  divisionName: string;
  status: 'Draft' | 'In Progress' | 'Completed';
  winnerTeamId?: string;
  winnerTeamName?: string;
  teamIds?: string[];
  sponsorIds?: string[];
  winnerTeamLogoUrl?: string;
}

export interface PlayerMatchPerformance {
  opponent: string;
  date: Date;
  matchId: string;
  runsScored?: number;
  ballsFaced?: number;
  battingStatus?: string;
  oversBowled?: number;
  runsConceded?: number;
  wicketsTaken?: number;
}

export interface Sponsor {
  sponsorId: string;
  name: string;
  logoUrl: string;
  website?: string;
}

export interface Transaction {
  transactionId: string;
  description: string;
  amount: number;
  type: 'Income' | 'Expense';
  category: 'Registration Fee' | 'Sponsorship' | 'Venue Hire' | 'Equipment' | 'Umpire Fees' | 'Other';
  date: Date;
}

export interface EquipmentItem {
  itemId: string;
  name: string;
  type: 'Bat' | 'Pads' | 'Gloves' | 'Helmet' | 'Ball' | 'Other';
  size?: string;
  status: 'Available' | 'Assigned' | 'Maintenance';
  currentAssignmentId?: string;
  currentHolderId?: string;
  currentHolderName?: string;
}

export interface FullEquipmentAssignment {
  assignmentId: string;
  itemId: string;
  itemName: string;
  itemType: EquipmentItem['type'];
  personId: string;
  personName: string;
  assignedDate: Date;
  returnedDate?: Date;
}

export interface Drill {
  drillId: string;
  name: string;
  description: string;
  category: 'Batting' | 'Bowling' | 'Fielding' | 'Fitness' | 'Tactical';
  duration: number; // in minutes
  userId?: string;
}

export interface TrainingSession {
  sessionId: string;
  title: string;
  date: Date;
  teamId: string;
  teamName: string;
  focus: string[];
  drills: {
    drillId: string;
    name: string;
    duration: number;
  }[];
  attendance: { personId: string; attended: boolean }[];
  notes?: string;
  userId?: string;
}

export interface AssignmentRequest {
  requestId: string;
  requesterId: string;
  requesterName: string;
  targetId: string;
  targetName: string;
  targetType: 'School' | 'Team';
  role: string;
  status: 'pending' | 'approved' | 'denied';
  createdAt: Date;
  reviewedBy?: string;
  reviewedByName?: string;
  reviewedAt?: Date;
  userId: string;
}

export interface PlayerPerformanceForecast {
  predictedPerformance: string;
  justification: string;
}

export interface HighlightReelOutput {
    highlights: {
        over: string;
        description: string;
        imageUrl: string;
    }[];
}


// From Player Tracker Spec
export interface PerformanceEntry {
  entryId: string;
  date: Date;
  matchId: string;
  runs: number;
  ballsFaced: number;
  wickets: number;
  oversBowled: number;
  wagonWheel: ShotData[];
  strikeRate: number;
  economyRate: number;
}

export interface SkillRating {
  ratingId: string;
  date: Date;
  batting: number;
  bowling: number;
  fielding: number;
  fitness: number;
  coachNotes?: string;
}

export interface TrainingLog {
  logId: string;
  date: Date;
  drillType: string;
  durationMins: number;
  coachNotes?: string;
}

export interface InjuryRecord {
  recordId: string;
  injuryType: string;
  startDate: Date;
  endDate?: Date;
  severity: 'Minor' | 'Moderate' | 'Severe';
  rehabNotes?: string;
  status: 'Active' | 'Recovered';
}

export interface Availability {
  availId: string;
  startDate: Date;
  endDate: Date;
  reason: string; // e.g., 'Injury', 'Personal', 'School Exam'
  status: 'Unavailable';
}

export interface Milestone {
  milestoneId: string;
  name: string; // e.g., '50th Match', '1000 Career Runs'
  achievedDate: Date;
}
```

```
export interface RunMapData {
    fineLeg: number;
    squareLeg: number;
    midWicket: number;
    longOn: number;
    cover: number;
    point: number;
    thirdMan: number;
    longOff: number;
}

export interface PlayerTrackerData {
  performanceEntries: PerformanceEntry[];
  skillRatings: SkillRating[];
  trainingLogs: TrainingLog[];
  injuryRecords: InjuryRecord[];
  availability: Availability[];
  milestones: Milestone[];
  runMap: RunMapData;
}

export interface AwardsData {
    trophyCabinet: Competition[];
    topRunScorer: LeaderboardPlayer | null;
    topWicketTaker: LeaderboardPlayer | null;
}

export interface AuditLog {
  logId: string;
  timestamp: Date;
  actorId: string;
  actorName: string;
  action: string; // e.g., 'user.role.update', 'match.delete'
  target: {
    type: string; // e.g., 'Person', 'Match'
    id: string;
    name?: string;
  };
  details?: Record<string, any>;
}

// Billing types
export interface InvoiceLineItem {
    id: string;
    description: string;
    quantity: number;
    unitPrice: number;
    total: number;
}

export interface Invoice {
    invoiceId: string;
    invoiceNumber: string;
    clientId: string;
    clientName: string;
    issueDate: Date;
    dueDate: Date;
    status: 'Draft' | 'Sent' | 'Paid' | 'Overdue';
    lineItems: InvoiceLineItem[];
    subtotal: number;
    tax: number;
    total: number;
    notes?: string;
}

export interface LiveMatchUpdateOutput {
    winProbability: number;
    summary: string;
    tacticalSuggestions?: string[];
}
```