

AI TABLE - Техническое задание

Платформа автоматизации Excel для МСФО

Версия документа: 1.0

Дата создания: Октябрь 2025

1. Общее описание проекта

1.1 Назначение системы

AI TABLE представляет собой веб-платформу для автоматизации финансовых расчетов в Excel файлах в соответствии с требованиями Международных стандартов финансовой отчетности (МСФО) и законодательством Республики Узбекистан.

1.2 Правовая база

Система создается в соответствии с требованиями:

- Постановления Президента РУз №ПП-4611 от 24.02.2020
- Постановления Президента РУз №ПП-282 от 15.09.2025
- Международными стандартами финансовой отчетности (IFRS)

1.3 Целевая аудитория

- Акционерные общества
- Коммерческие банки
- Страховые организации
- Крупные налогоплательщики
- Общественно значимые организации (PIE)
- Финансовые специалисты и бухгалтеры

1.4 Бизнес-цели

- Автоматизация расчетов амортизации, скидок и обесценения
 - Обеспечение соответствия требованиям МСФО
 - Минимизация ошибок в финансовой отчетности
 - Сокращение времени подготовки отчетности
-

2. Функциональные требования

2.1 Пользовательские роли

2.1.1 Неавторизованный пользователь

Права доступа:

- Просмотр лендинговой страницы
- Ознакомление с функциональностью
- Инициация процесса регистрации

Ограничения:

- Отсутствие доступа к функциональности обработки файлов
- Невозможность сохранения результатов

2.1.2 Авторизованный пользователь (Basic)**Права доступа:**

- Загрузка Excel файлов (до 5 файлов в месяц)
- Применение базовых МСФО шаблонов
- Просмотр результатов обработки
- Экспорт обработанных файлов
- Просмотр истории обработки (последние 10 файлов)

Ограничения:

- Лимит на количество обрабатываемых файлов
- Ограниченный набор шаблонов
- Отсутствие приоритетной обработки

2.1.3 Авторизованный пользователь (Premium)**Права доступа:**

- Неограниченная загрузка файлов
- Доступ ко всем МСФО шаблонам
- Приоритетная обработка
- Полная история обработки
- Техническая поддержка
- API доступ

2.2 Основные функциональные модули**2.2.1 Модуль аутентификации****Требования:**

- Интеграция с Google OAuth 2.0
- Автоматическое создание профиля пользователя
- Управление сессиями через JWT токены
- Хранение токенов в Redis с TTL
- Обновление токенов (refresh token flow)

Атрибуты пользователя:

```
{
  googleId: String,
  email: String,
  name: String,
  picture: String,
  subscriptionType: String, // 'basic' | 'premium'
  createdAt: Date,
  lastLoginAt: Date,
  isActive: Boolean
}
```

2.2.2 Модуль управления шаблонами

Типы шаблонов:

1. Амортизация основных средств

- Линейный метод
- Метод уменьшающегося остатка
- Метод суммы лет
- Производственный метод

2. Расчеты скидок

- Торговые скидки
- Скидки за досрочную оплату
- Сезонные скидки
- Объемные скидки

3. Обесценение активов

- Тестирование на обесценение
- Расчет возмещаемой стоимости
- Признание убытков от обесценения

4. Финансовые отчеты

- Отчет о прибылях и убытках
- Отчет о финансовом положении
- Отчет о движении денежных средств

Структура шаблона:

```
{
  id: String,
  name: String,
  category: String,
  description: String,
  ifrsStandard: String, // IAS 16, IFRS 9, etc.
  formulas: [{
```

```
    cellRange: String,  
    formula: String,  
    validation: Object  
  }],  
  requiredFields: [String],  
  outputFormat: Object,  
  isActive: Boolean,  
  version: String  
}
```

2.2.3 Модуль обработки файлов

Поддерживаемые форматы:

- Excel (.xlsx, .xls)
- CSV (с автоматическим определением разделителей)

Максимальные ограничения:

- Размер файла: 50 МБ (Basic), 200 МБ (Premium)
- Количество строк: 100,000
- Количество столбцов: 500

Процесс обработки:

1. Валидация файла

- Проверка формата
- Проверка размера
- Сканирование на вредоносный код
- Проверка структуры данных

2. Анализ содержимого

- Определение типа данных в ячейках
- Идентификация финансовых показателей
- Поиск существующих формул
- Анализ структуры таблицы

3. Применение шаблонов

- Сопоставление данных с шаблонами
- Применение МСФО формул
- Валидация результатов
- Генерация отчета об изменениях

4. Генерация результата

- Создание новой версии файла
- Сохранение метаданных обработки
- Подготовка файла для скачивания

API обработки:

```
POST /api/files/process
Content-Type: multipart/form-data

{
  file: File,
  templateId: String,
  options: {
    preserveFormatting: Boolean,
    addComments: Boolean,
    generateReport: Boolean
  }
}

Response:
{
  processId: String,
  status: 'processing' | 'completed' | 'error',
  originalFile: {
    name: String,
    size: Number,
    sheets: [String]
  },
  processedFile: {
    url: String,
    expiresAt: Date
  },
  changes: [{
    sheet: String,
    cell: String,
    oldValue: Any,
    newValue: Any,
    formulaApplied: String
  }],
  report: {
    summary: String,
    warnings: [String],
    errors: [String]
  }
}
```

2.2.4 Модуль сравнения результатов**Функциональность:**

- Отображение исходного файла (read-only)
- Отображение обработанного файла
- Подсветка измененных ячеек
- Детализация примененных формул

- Экспорт отчета о изменениях

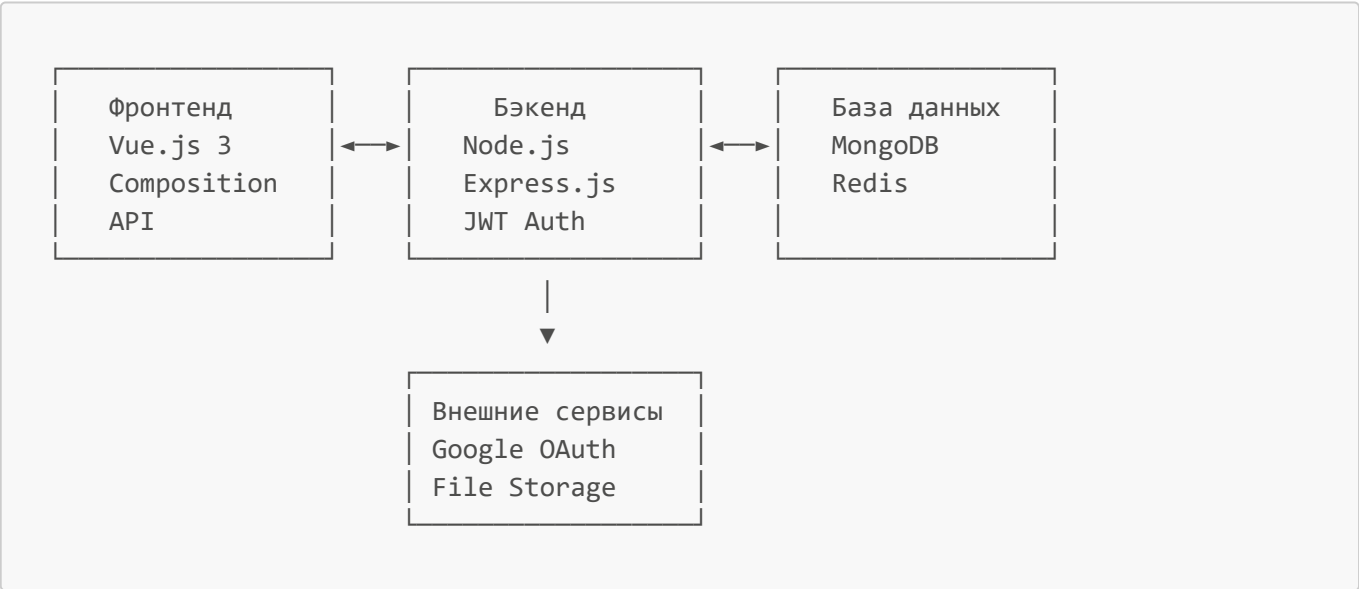
Интерфейс сравнения:

```
<template>
  <div class="comparison-view">
    <div class="tabs">
      <Tab name="before" label="До обработки" />
      <Tab name="after" label="После обработки" />
      <Tab name="changes" label="Изменения" />
    </div>

    <div class="spreadsheet-container">
      <SpreadsheetViewer
        :data="currentData"
        :readonly="currentTab === 'before'"
        :highlights="changes"
      />
    </div>
  </div>
</template>
```

3. Техническая архитектура

3.1 Общая архитектура системы



3.2 Бэкенд архитектура

3.2.1 Структура проекта

```
src/
├── config/
```

```
| database.js
| redis.js
| oauth.js
| constants.js
| components/
| | auth/
| | | controller.js
| | | routes.js
| | | middleware.js
| | | model.js
| | users/
| | | controller.js
| | | routes.js
| | | middleware.js
| | | model.js
| | files/
| | | controller.js
| | | routes.js
| | | middleware.js
| | | model.js
| | templates/
| | | controller.js
| | | routes.js
| | | middleware.js
| | | model.js
| middleware/
| | validation.js
| | upload.js
| | rateLimit.js
| services/
| | auth.js
| | file.js
| | excel.js
| | template.js
| utils/
| | logger.js
| | errors.js
| | validators.js
| app.js
```

3.2.2 Основные зависимости

```
{
  "engines": {
    "node": "24.x"
  },
  "dependencies": {
    "express": "^4.18.2",
    "mongoose": "^7.5.0",
    "redis": "^4.6.8",
    "jsonwebtoken": "^9.0.2",
```

```

    "passport": "^0.6.0",
    "passport-google-oauth20": "^2.0.0",
    "multer": "^1.4.5",
    "exceljs": "^4.3.0",
    "helmet": "^7.0.0",
    "cors": "^2.8.5",
    "joi": "^17.9.2",
    "bcrypt": "^5.1.1",
    "express-rate-limit": "^6.8.1",
    "winston": "^3.10.0"
  }
}

```

3.2.3 API Routes

Аутентификация:

```

GET  /api/auth/google           // Инициация OAuth
GET  /api/auth/google/callback  // Callback OAuth
POST /api/auth/refresh          // Обновление токена
POST /api/auth/logout           // Выход
GET  /api/auth/profile           // Профиль пользователя

```

Файлы:

```

POST /api/files/upload           // Загрузка файла
POST /api/files/process          // Обработка файла
GET  /api/files/download/:id     // Скачивание результата
GET  /api/files/history          // История обработки
DELETE /api/files/:id            // Удаление файла
GET  /api/files/status/:jobId    // Статус обработки

```

Шаблоны:

```

GET  /api/templates              // Список доступных шаблонов
GET  /api/templates/:id          // Детали шаблона
POST /api/templates/validate     // Валидация данных для шаблона

```

3.3 Фронтенд архитектура

3.3.1 Структура проекта

```

src/
├── components/

```



```
├── common/
│   ├── Header.vue
│   ├── Footer.vue
│   └── LoadingSpinner.vue
├── auth/
│   ├── LoginButton.vue
│   └── UserProfile.vue
├── files/
│   ├── FileUpload.vue
│   ├── ProcessingStatus.vue
│   └── FileHistory.vue
├── spreadsheet/
│   ├── SpreadsheetViewer.vue
│   ├── ComparisonView.vue
│   └── CellEditor.vue
├── views/
│   ├── LandingPage.vue
│   ├── Dashboard.vue
│   ├── ProcessingPage.vue
│   └── ResultsPage.vue
├── stores/
│   ├── auth.js
│   ├── files.js
│   └── templates.js
├── services/
│   ├── api.js
│   ├── auth.service.js
│   └── file.service.js
├── utils/
│   ├── constants.js
│   └── validators.js
├── router/
│   └── index.js
```

3.2.3 Основные зависимости фронтенда

```
{
  "engines": {
    "node": "24.x"
  },
  "dependencies": {
    "vue": "^3.3.4",
    "vue-router": "^4.2.4",
    "pinia": "^2.1.6",
    "axios": "^1.5.0",
    "@vueuse/core": "^10.4.1",
    "lucide-vue-next": "^0.279.0",
    "tailwindcss": "^3.3.3"
  }
}
```

4. База данных

4.1 MongoDB коллекции

4.1.1 Коллекция Users

```
{
  _id: ObjectId,
  googleId: String, // unique
  email: String, // unique
  name: String,
  picture: String,
  subscriptionType: {
    type: String,
    enum: ['basic', 'premium'],
    default: 'basic'
  },
  subscriptionExpiresAt: Date,
  filesProcessedThisMonth: {
    type: Number,
    default: 0
  },
  monthlyResetDate: Date,
  isActive: {
    type: Boolean,
    default: true
  },
  preferences: {
    language: {
      type: String,
      enum: ['ru', 'uz', 'en'],
      default: 'ru'
    },
    defaultTemplate: String,
    notifications: {
      email: Boolean,
      browser: Boolean
    }
  },
  createdAt: Date,
  updatedAt: Date,
  lastLoginAt: Date
}
```

4.1.2 Коллекция Templates

```
{
  _id: ObjectId,
```

```

name: {
  ru: String,
  uz: String,
  en: String
},
description: {
  ru: String,
  uz: String,
  en: String
},
category: {
  type: String,
  enum: ['depreciation', 'discounts', 'impairment', 'reports']
},
ifrsStandard: String, // IAS 16, IFRS 9, etc.
subscriptionRequired: {
  type: String,
  enum: ['basic', 'premium']
},
formulas: [{
  name: String,
  description: String,
  cellPattern: String, // Regular expression for cell matching
  formula: String,
  validation: {
    required: Boolean,
    type: String, // 'number', 'date', 'text'
    min: Number,
    max: Number,
    format: String
  }
}],
requiredColumns: [String],
outputFormat: {
  addColumns: [String],
  modifications: Object
},
isActive: Boolean,
version: String,
createdAt: Date,
updatedAt: Date
}

```

4.1.3 Коллекция ProcessingJobs

```

{
  _id: ObjectId,
  userId: ObjectId,
  originalFileName: String,
  processedFileName: String,
  templateId: ObjectId,

```

```

status: {
  type: String,
  enum: ['pending', 'processing', 'completed', 'failed']
},
fileSize: Number,
originalFileUrl: String,
processedFileUrl: String,
expiresAt: Date,
processingOptions: {
  preserveFormatting: Boolean,
  addComments: Boolean,
  generateReport: Boolean
},
changes: [{
  sheet: String,
  cellAddress: String,
  oldValue: Schema.Types.Mixed,
  newValue: Schema.Types.Mixed,
  formulaApplied: String,
  changeType: {
    type: String,
    enum: ['formula', 'value', 'format']
  }
}],
processingReport: {
  summary: String,
  totalChanges: Number,
  warnings: [String],
  errors: [String],
  processingTime: Number // milliseconds
},
error: {
  message: String,
  stack: String,
  code: String
},
createdAt: Date,
completedAt: Date
}

```

4.1.4 Коллекция FileHistory

```

{
  _id: ObjectId,
  userId: ObjectId,
  fileName: String,
  templateName: String,
  processingJobId: ObjectId,
  fileSize: Number,
  status: String,
  downloadCount: {

```

```
    type: Number,  
    default: 0  
  },  
  lastDownloadAt: Date,  
  tags: [String], // User tags  
  notes: String, // User notes  
  createdAt: Date  
}
```

4.2 Redis структуры

4.2.1 Сессии пользователей

```
Key: session:{userId}  
Value: {  
  accessToken: String,  
  refreshToken: String,  
  expiresAt: Number,  
  userAgent: String,  
  ipAddress: String  
}  
TTL: 24 hours
```

4.2.2 Rate limiting

```
Key: rateLimit:{userId}:{endpoint}  
Value: Number (request count)  
TTL: 1 hour
```

4.2.3 File processing status

```
Key: processing:{jobId}  
Value: {  
  status: String,  
  progress: Number,  
  currentStep: String,  
  eta: Number  
}  
TTL: 1 hour
```

5. UI/UX требования

5.1 Дизайн-система

5.1.1 Цветовая палитра

```
:root {  
  --primary: #765fc6;      /* Violet primary */  
  --primary-light: #8b7ed8; /* Lighter violet */  
  --primary-dark: #5d4a9e;  /* Darker violet */  
  --secondary: #10B981;     /* Emerald-500 */  
  --danger: #EF4444;        /* Red-500 */  
  --warning: #F59E0B;       /* Amber-500 */  
  --success: #10B981;       /* Emerald-500 */  
  --gray-50: #F9FAFB;  
  --gray-100: #F3F4F6;  
  --gray-500: #6B7280;  
  --gray-900: #111827;  
  --accent: #9333ea;        /* Purple accent */  
}
```

5.1.2 Типографика

```
/* Headers */  
.h1 { font-size: 2.25rem; font-weight: 700; font-family: 'Montserrat', sans-serif;  
}  
.h2 { font-size: 1.875rem; font-weight: 600; font-family: 'Montserrat', sans-  
  serif; }  
.h3 { font-size: 1.5rem; font-weight: 600; font-family: 'Montserrat', sans-serif;  
}  
  
/* Body text */  
.body { font-size: 1rem; line-height: 1.5; font-family: 'Montserrat', sans-serif;  
}  
.small { font-size: 0.875rem; line-height: 1.25; font-family: 'Montserrat', sans-  
  serif; }
```

5.2 Адаптивность

- Мобильные устройства: 320px - 768px
- Планшеты: 768px - 1024px
- Десктоп: 1024px+

5.3 Доступность

- Соответствие WCAG 2.1 AA
- Поддержка навигации с клавиатуры
- Альтернативный текст для изображений
- Высокий контраст цветов

6. Безопасность

6.1 Аутентификация и авторизация

- OAuth 2.0 с Google
- JWT токены с коротким временем жизни (15 минут)
- Refresh токены с длительным временем жизни (7 дней)
- Валидация токенов на каждом запросе

6.2 Защита от атак

```
// Rate limiting
const rateLimit = require('express-rate-limit');

const limiter = rateLimit({
  windowMs: 15 * 60 * 1000, // 15 minutes
  max: 100, // limit requests per IP
  message: 'Too many requests from this IP address'
});
```

6.3 Валидация данных

```
const Joi = require('joi');

const fileUploadSchema = Joi.object({
  templateId: Joi.string().required(),
  options: Joi.object({
    preserveFormatting: Joi.boolean().default(true),
    addComments: Joi.boolean().default(false),
    generateReport: Joi.boolean().default(true)
  })
});
```

6.4 Защита файлов

- Сканирование загружаемых файлов на вирусы
- Ограничение типов файлов
- Временные ссылки для скачивания (expires через 24 часа)
- Шифрование файлов при хранении

7. Производительность

7.1 Требования к производительности

- Время загрузки главной страницы: < 2 сек
- Время обработки файла (до 10 МБ): < 30 сек

- Время ответа API: < 500 мс
- Поддержка 1000+ одновременных пользователей

7.2 Оптимизации

7.2.1 Кэширование

```
// Cache templates with Redis
const getTemplate = async (templateId) => {
  const cached = await redis.get(`template:${templateId}`);
  if (cached) return JSON.parse(cached);

  const template = await Template.findById(templateId);
  await redis.setex(`template:${templateId}`, 3600, JSON.stringify(template));
  return template;
};
```

7.2.2 Asynchronous file processing

```
const processFileAsync = async (jobId, file, template) => {
  try {
    await updateJobStatus(jobId, 'processing');
    const result = await applyTemplate(file, template);
    await updateJobStatus(jobId, 'completed', result);
  } catch (error) {
    await updateJobStatus(jobId, 'failed', null, error);
  }
};
```

8. Развертывание и DevOps

8.1 Контейнеризация

8.1.1 Dockerfile для бэкенда

```
FROM node:24-alpine

WORKDIR /app

COPY package*.json ./
RUN npm ci --only=production

COPY . .

EXPOSE 3000
```



```
CMD ["node", "src/app.js"]
```

8.1.2 Docker Compose

```
version: '3.8'

services:
  app:
    build: .
    ports:
      - "3000:3000"
    environment:
      - NODE_ENV=production
      - MONGODB_URI=mongodb://mongo:27017/aitable
      - REDIS_URI=redis://redis:6379
    depends_on:
      - mongo
      - redis

  mongo:
    image: mongo:6.0
    volumes:
      - mongo_data:/data/db

  redis:
    image: redis:7-alpine
    volumes:
      - redis_data:/data

volumes:
  mongo_data:
  redis_data:
```

8.2 CI/CD Pipeline

```
# .gitlab-ci.yml
# Include the universal template for deployment
include:
  - project: 'templates/gitlab-ci'
    file: '.gitlab-ci-template.yml'

stages:
  - build
  - deploy

variables:
  NODE_VERSION: "24"
```

```
build:
  stage: build
  image: node:24-alpine
  script:
    - npm ci
    - npm run build
  artifacts:
    paths:
      - dist/
    expire_in: 1 hour

deploy_production:
  stage: deploy
  only:
    - main
  script:
    - chmod +x deploy.sh
    - ./deploy.sh
  environment:
    name: production
    url: https://ai-table.com
```

9. Мониторинг и логирование

9.1 Логирование

```
const winston = require('winston');

const logger = winston.createLogger({
  level: 'info',
  format: winston.format.combine(
    winston.format.timestamp(),
    winston.format.errors({ stack: true }),
    winston.format.json()
  ),
  transports: [
    new winston.transports.File({ filename: 'error.log', level: 'error' }),
    new winston.transports.File({ filename: 'combined.log' })
  ]
});
```

9.2 Метрики производительности

- Время ответа API endpoints
- Количество обработанных файлов
- Использование памяти и CPU
- Количество активных пользователей
- Ошибки и их частота

10. Планы развития

10.1 Версия 1.0 (MVP)

- ☒ Базовая аутентификация через Google
- ☒ Загрузка и обработка Excel файлов
- ☒ 4 основных типа шаблонов МСФО
- ☒ Интерфейс сравнения результатов
- ☒ Базовая система подписок

10.2 Версия 1.1

- ☐ API для интеграции с внешними системами
- ☐ Пакетная обработка файлов
- ☐ Расширенная аналитика использования
- ☐ Многоязычная поддержка (узбекский, английский)

10.3 Версия 2.0

- ☐ ИИ-помощник для консультаций по МСФО
- ☐ Интеграция с популярными учетными системами
- ☐ Мобильное приложение
- ☐ Корпоративные функции (команды, роли, права)

11. Заключение

Данное техническое задание определяет полный объем работ по созданию платформы AI TABLE для автоматизации МСФО расчетов в Excel. Документ содержит детальные спецификации всех компонентов системы и может служить основой для разработки, тестирования и развертывания продукта.

Ключевые принципы реализации:

- Модульная архитектура для легкости расширения
- Безопасность данных как приоритет
- Производительность и масштабируемость
- Соответствие международным стандартам МСФО
- Простота использования для конечных пользователей

Критерии готовности к релизу:

- Время обработки файлов в заданных лимитах
- Успешное прохождение нагрузочного тестирования
- Соответствие требованиям безопасности
- Валидация МСФО расчетов экспертами

Версия документа: 1.0

Дата последнего обновления: Октябрь 2025