

Pickabook Personalisation Prototype – Complete Repository Files

Below is the **entire folder structure** and **all required repo files** so you can copy-paste them into GitHub.

Project Structure

```
pickabook-prototype/
├── frontend/
│   ├── package.json
│   ├── next.config.js
│   ├── tailwind.config.js
│   ├── postcss.config.js
│   ├── app/
│   │   └── page.js
│   ├── components/
│   │   └── UploadBox.jsx
│   └── styles/
│       └── globals.css
├── backend/
│   ├── package.json
│   ├── server.js
│   ├── controllers/
│   │   └── personalizeController.js
│   └── utils/
│       └── mockAI.js
├── README.md
├── .gitignore
└── .env.example
```



FRONTEND FILES (Next.js)

frontend/package.json

```
{
  "name": "pickabook-frontend",
  "version": "1.0.0",
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start"
  },
  "dependencies": {
    "next": "14.0.0",
    "react": "18.2.0",
    "react-dom": "18.2.0"
  }
}
```

frontend/next.config.js

```
module.exports = {
  reactStrictMode: true,
};
```

frontend/tailwind.config.js

```
module.exports = {
  content: ["/app/**/*.{js,jsx}", "/components/**/*.{js,jsx}"],
  theme: { extend: {} },
  plugins: [],
};
```

frontend/postcss.config.js

```
module.exports = {
  plugins: {
    tailwindcss: {},
    autoprefixer: {},
  },
}
```

```
  },  
};
```

frontend/styles/globals.css

```
@tailwind base;  
@tailwind components;  
@tailwind utilities;
```

frontend/components/UploadBox.jsx

```
'use client';  
import { useState } from 'react';  
  
export default function UploadBox() {  
  const [image, setImage] = useState(null);  
  const [result, setResult] = useState(null);  
  
  const uploadImage = async () => {  
    const formData = new FormData();  
    formData.append('file', image);  
  
    const res = await fetch('http://localhost:5000/api/personalize', {  
      method: 'POST',  
      body: formData  
    });  
  
    const data = await res.json();  
    setResult(data.outputImage);  
  };  
  
  return (  
    <div className="p-6 border rounded-xl w-full max-w-lg mx-auto">  
      <h2 className="text-xl font-bold mb-4">Upload Child Photo</h2>  
  
      <input  
        type="file"  
        onChange={(e) => setImage(e.target.files[0])}  
        className="mb-4"  
      />  
  
      <button  
        onClick={uploadImage}  
        className="px-4 py-2 bg-blue-600 text-white rounded-lg"
```

```

    >
      Personalize
    </button>

    {result && (
      <img src={result} alt="Result" className="mt-4 rounded-lg" />
    )}
  </div>
);
}

```

frontend/app/page.js

```

import UploadBox from "../components/UploadBox";

export default function Home() {
  return (
    <main className="p-10 flex justify-center">
      <UploadBox />
    </main>
  );
}

```

BACKEND FILES (Node.js + Express)

backend/package.json

```

{
  "name": "pickabook-backend",
  "version": "1.0.0",
  "scripts": {
    "start": "node server.js"
  },
  "dependencies": {
    "cors": "^2.8.5",
    "express": "^4.18.2",
    "multer": "^1.4.5"
  }
}

```

backend/server.js

```
const express = require('express');
const multer = require('multer');
const cors = require('cors');
const personalize = require('./controllers/personalizeController');

const app = express();
app.use(cors());

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

app.post('/api/personalize', upload.single('file'), personalize);

app.listen(5000, () => console.log('Backend running on port 5000'));
```

backend/controllers/personalizeController.js

```
const mockAI = require('../utils/mockAI');

module.exports = async (req, res) => {
  const filePath = req.file.path;

  const output = await mockAI(filePath);

  res.json({ outputImage: output });
};
```

backend/utils/mockAI.js

```
module.exports = async function mockAI(inputImage) {
  return "https://placeholder.co/400x400?text=Mock+AI+Generated+Image";
};
```



README.md

Pickabook Personalisation Prototype

A simple end-to-end prototype where users upload a child photo and the backend

```
returns a mock personalised illustration.
```

```
## 🧰 Tech Stack
- Next.js (frontend)
- Node.js + Express (backend)
- Multer for file uploads
```

```
## 📦 Setup
```

```
git clone <repo> cd frontend && npm install cd ../backend && npm install
```

```
---
## ▶ Run Backend
```

```
cd backend node server.js
```

```
## ▶ Run Frontend
```

```
cd frontend npm run dev
```

```
Frontend → `http://localhost:3000`
Backend → `http://localhost:5000`
```

.env.example

```
API_KEY=your_api_key_here
AI_MODEL=mock
```

Your complete repo is ready!

You can now push ALL of these files to GitHub.