Complete GitHub Upload Guide

□ Step-by-Step Guide to Upload Your Project to GitHub

PART 1: Prepare Your Project

Step 1: Create `.gitignore` Files

Why? To prevent uploading large folders and sensitive files.

Create `.gitignore` in SERVER folder

Location: `server/.gitignore`

Content:

```
# Dependencies
node_modules/

# Environment variables
.env

# Logs
*.log
npm-debug.log*

# OS files
.DS_Store
Thumbs.db

# IDE
.vscode/
.idea/
```

Create `.gitignore` in CLIENT folder

Location: `client/.gitignore`

Content:

```
# Dependencies
node_modules/

# Production build
build/
dist/

# Testing
coverage/

# Misc
.DS_Store
.env.local
.env.development.local
.env.test.local
.env.production.local

# Logs
npm-debug.log*
yarn-debug.log*
yarn-error.log*

# IDE
.vscode/
.idea/
```

Create `.gitignore` in ROOT folder

Location: `edtech-task-manager/.gitignore` (main project folder)

Content:

```
# Dependencies
node_modules/

# Environment variables
.env

# Logs
*.log

# OS
.DS_Store
```

Step 2: Create README.md

Location: `edtech-task-manager/README.md`

Copy the README I provided earlier (from previous artifact).

Step 3: Create `.env.example` (Template)

Location: `server/.env.example`

Content:

```
PORT=5000
MONGODB_URI=mongodb+srv://username:password@cluster.mongodb.net/edtech-task-manager?retryWrites=true&w=majority
JWT_SECRET=your-super-secret-jwt-key-minimum-32-characters-long
NODE_ENV=development
```

⚠ **Important:** This is a template WITHOUT actual credentials!

PART 2: Install Git (If Not Already Installed)

Check if Git is Installed

Open Command Prompt and type:

```
git --version
```

If you see: `git version 2.x.x` → Git is installed □

If you see: `'git' is not recognized` → Install Git

Install Git

1. Go to: https://git-scm.com/download/win (https://git-scm.com/download/win)
2. Download Windows installer
3. Run installer → Keep clicking "Next" (default settings are fine)
4. Restart Command Prompt after installation

PART 3: Create GitHub Repository

Step 1: Sign up/Login to GitHub

1. Go to: https://github.com (https://github.com)
2. Sign up (if you don't have an account) or Login

Step 2: Create New Repository

1. Click the **"+"** icon (top right) → **"New repository"**
2. Fill in:
   - **Repository name:** `edtech-task-manager`
   - **Description:** `Full-stack EdTech learning task manager with role-based access`
   - **Public** or **Private** (choose Public for assignment)
   - □ **DO NOT** check "Add a README file"
   - □ **DO NOT** check "Add .gitignore"
   - □ **DO NOT** choose a license
3. Click **"Create repository"**

Step 3: Copy Repository URL

You'll see a page with commands. Copy the **HTTPS URL**, looks like:

```
https://github.com/yourusername/edtech-task-manager.git
```

PART 4: Upload Your Project

Step 1: Open Terminal in VS Code

1. Open your project in VS Code
2. Open Terminal: **Terminal → New Terminal**
3. Make sure you're in the **main project folder**

Check current location:

```
cd
```

Should show: `C:\Users\latha\OneDrive\Desktop\edtech-task-manager`

Step 2: Initialize Git Repository

Type these commands ONE BY ONE:

```
git init
```

Expected output:

```
Initialized empty Git repository in C:/Users/latha/OneDrive/Desktop/edtech-task-manager/.git/
```

Step 3: Configure Git (First Time Only)

Set your name and email:

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

Example:

```
git config --global user.name "Latha Kumar"
git config --global user.email "latha@example.com"
```

Step 4: Add All Files to Git

```
git add .
```

The dot (.) means "add all files"

Step 5: Check What Will Be Committed

```
git status
```

You should see:

```
On branch master

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md
        new file:   client/package.json
        new file:   client/src/App.js
        new file:   server/package.json
        new file:   server/src/server.js
        ... (many more files)
```

 Good signs:

- You see `README.md`
- You see files from `client/` and `server/`
- You do **NOT** see `node_modules/` (should be ignored)
- You do **NOT** see `.env` (should be ignored)

 If you see `node_modules/` or `.env`:

- Make sure `.gitignore` files are created correctly
- Run: `git rm -r --cached node_modules`
- Run: `git rm --cached server/.env`
- Then: `git add .` again

Step 6: Commit Your Code

```
git commit -m "Initial commit: EdTech Task Manager with role-based access"
```

Expected output:

```
[master (root-commit) abc1234] Initial commit: EdTech Task Manager with role-based access
 150 files changed, 15000 insertions(+)
 create mode 100644 README.md
 create mode 100644 client/package.json
 ...
```

Step 7: Rename Branch to Main (Optional but Recommended)

```
git branch -M main
```

Step 8: Add GitHub Repository as Remote

Replace with YOUR repository URL:

```
git remote add origin https://github.com/yourusername/edtech-task-manager.git
```

Example:

```
git remote add origin https://github.com/lathakumar/edtech-task-manager.git
```

Step 9: Push to GitHub

```
git push -u origin main
```

You may be asked to login to GitHub:

1. A browser window will open
2. Click "Authorize Git Credential Manager"
3. Login to GitHub if needed

Expected output:

```
Enumerating objects: 200, done.
Counting objects: 100% (200/200), done.
Delta compression using up to 8 threads
Compressing objects: 100% (180/180), done.
Writing objects: 100% (200/200), 1.50 MiB | 500.00 KiB/s, done.
Total 200 (delta 50), reused 0 (delta 0)
To https://github.com/yourusername/edtech-task-manager.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

 Success! Your code is now on GitHub!

PART 5: Verify Upload

Step 1: Check GitHub Website

1. Go to: `https://github.com/yourusername/edtech-task-manager`
2. You should see:
   - README.md displaying nicely
   - `client/` folder
   - `server/` folder
   - `.gitignore` files
   - **No** `node_modules/` folders
   - **No** `.env` file

Step 2: Check File Count

You should see something like:

```
150+ files, ~15,000 lines of code
```

PART 6: Make Additional Commits (Show Development Process)

Why Multiple Commits?

The assignment requires **"Multiple meaningful commits showing your development process"**

Create Meaningful Commits

Option 1: Separate by Feature (Recommended)

```
# Commit 1: Backend authentication
git add server/src/middleware/auth.js server/src/routes/auth.js server/src/controllers/authController.js
git commit -m "feat: Implement JWT authentication and user signup/login"
git push

# Commit 2: Task management
git add server/src/routes/tasks.js server/src/controllers/taskController.js
git commit -m "feat: Add task CRUD operations with role-based access"
git push

# Commit 3: Frontend components
git add client/src/components/
git commit -m "feat: Build React components for auth and dashboard"
git push

# Commit 4: Role-based UI
git add client/src/components/Dashboard/Dashboard.jsx
git commit -m "feat: Implement role-based task filtering and ownership checks"
git push

# Commit 5: Documentation
git add README.md
git commit -m "docs: Add comprehensive README with setup instructions"
git push
```

Option 2: Create Commits Retroactively

If you already pushed everything in one commit, you can create additional meaningful commits:

```
# Make a small change (add a comment in code)
# Open server/src/server.js and add a comment at the top:
# // EdTech Task Manager - Main Server File

git add server/src/server.js
git commit -m "docs: Add documentation comments to server file"
git push

# Make another small improvement
# Update a validation message or add error handling

git add .
git commit -m "fix: Improve error messages for better user experience"
git push
```

PART 7: Common Issues & Solutions

Issue 1: "fatal: not a git repository"

Solution:

```
git init
```

Issue 2: "error: src refspec main does not exist"

Solution:

```
git branch -M main
git push -u origin main
```

Issue 3: "! [rejected] main -> main (fetch first)"

Solution:

```
git pull origin main --allow-unrelated-histories
git push -u origin main
```

Issue 4: ".env file is in repository"

Solution:

```
# Remove .env from git tracking
git rm --cached server/.env

# Make sure .gitignore includes .env
echo ".env" >> .gitignore

# Commit the change
git add .gitignore
git commit -m "fix: Remove .env from repository"
git push
```

Issue 5: "node_modules uploaded to GitHub"

Solution:

```
# Remove node_modules from git
git rm -r --cached node_modules
git rm -r --cached client/node_modules
git rm -r --cached server/node_modules

# Make sure .gitignore is correct
git add .gitignore
git commit -m "fix: Remove node_modules from repository"
git push
```

PART 8: Final Checklist

Before submitting, verify on GitHub:

- [ ] □ README.md displays correctly
- [ ] □ All code files visible (client & server folders)
- [ ] □ `.gitignore` files present
- [ ] □ `.env.example` included (template without credentials)
- [ ] □ **No** actual `.env` file (with passwords)
- [ ] □ **No** `node_modules/` folders
- [ ] □ At least 3-5 meaningful commits
- [ ] □ Commit messages are descriptive
- [ ] □ Repository is Public (if required)

□ Quick Command Summary

Complete sequence for first-time upload:

```
# 1. Navigate to project folder
cd C:\Users\latha\OneDrive\Desktop\edtech-task-manager

# 2. Initialize git
git init

# 3. Configure git (first time only)
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"

# 4. Add all files
git add .

# 5. Create first commit
git commit -m "Initial commit: Complete EdTech Task Manager"

# 6. Rename branch
git branch -M main

# 7. Add remote repository
git remote add origin https://github.com/yourusername/edtech-task-manager.git

# 8. Push to GitHub
git push -u origin main
```

For additional commits later:

```
# Make changes to your files...

# Stage changes
git add .

# Commit with message
git commit -m "feat: Add new feature or fix bug"

# Push to GitHub
git push
```

□ Good Commit Message Examples

Format: `type: description`

Types:

- `feat:` - New feature
- `fix:` - Bug fix
- `docs:` - Documentation changes
- `style:` - Formatting, no code change
- `refactor:` - Code restructuring
- `test:` - Adding tests
- `chore:` - Maintenance tasks

Examples:

```
git commit -m "feat: Implement JWT authentication middleware"
git commit -m "feat: Add role-based task filtering for teachers"
git commit -m "fix: Correct task ownership validation logic"
git commit -m "docs: Update README with API documentation"
git commit -m "style: Improve dashboard UI with Tailwind CSS"
git commit -m "refactor: Optimize MongoDB queries for better performance"
git commit -m "fix: Handle edge case when student has no assigned teacher"
```

□ You're Done!

After completing these steps:

1. □ Your code is on GitHub
2. □ README is visible
3. □ Multiple commits showing development process
4. □ No sensitive files (.env, node_modules)
5. □ Ready to share repository link in your submission!

□ Email Submission Template

```
Subject: EdTech Task Manager - Full-Stack Developer Assignment Submission

Dear Hiring Team,

I am submitting my completed EdTech Learning Task Manager assignment.

Project Details:
- GitHub Repository: https://github.com/yourusername/edtech-task-manager
- Video Walkthrough: [YouTube/Google Drive Link]
- Live Demo (if deployed): [Render/Vercel Link - optional]

Key Features Implemented:
 Role-based access control (Student & Teacher)
 JWT authentication with bcrypt password hashing
 Full CRUD operations on tasks
 Task filtering by progress status
 Teacher-student relationship management
 Beautiful, responsive UI with Tailwind CSS
 Comprehensive error handling and validation

Tech Stack:
- Frontend: React, React Router, Axios
- Backend: Node.js, Express, MongoDB, Mongoose
- Authentication: JWT, bcrypt
- Validation: Joi

The README file contains:
- Complete setup instructions
- API documentation
- Database schema
- Security features explanation
- AI assistance disclosure

All requirements have been met and tested thoroughly.

Thank you for your consideration.

Best regards,
[Your Name]
[Your Email]
[Your Phone - optional]
```

Need help with:

- □ "Recording the video walkthrough?"
- □ "Deploying to Render (bonus)?"
- □ "Specific git errors?"

Let me know! You're almost at the finish line! □□